UNIVERSIDADE FEDERAL DO PARANÁ - UFPR
PROGRAMA DE PÓS-GRADUAÇÃO EM MÉTODOS NUMÉRICOS EM ENGENHARIA – PPGMNE

GENIVAL PAVANELLI

EXTRAÇÃO DE REGRAS DE CLASSIFICAÇÃO DE BASES DE DADOS POR MEIO DE PROCEDIMENTOS META-HEURÍSTICOS BASEADOS EM *GRASP*

CURITIBA
2014

GENIVAL PAVANELLI

EXTRAÇÃO DE REGRAS DE CLASSIFICAÇÃO DE BASES DE DADOS POR MEIO DE PROCEDIMENTOS META-HEURÍSTICOS BASEADOS EM *GRASP*

Tese apresentada ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia, na Área de Concentração em Programação Matemática, dos setores de Ciências Exatas e de Tecnologia da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Doutor.

Orientadora: Profa. Dra. Maria Teresinha Arns Steiner

CURITIBA
2014

P337e

Pavanelli, Genival

Extração de regras de classificação de bases de dados por meio de procedimentos meta-heurísticos baseados em GRASP / Genival Pavanelli Gomes Santos. – Curitiba, 2014.

130f. : il. color. ; 30 cm.

Tese (doutorado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-graduação em Métodos Numéricos em Engenharia, 2014.

Orientador: Maria Teresinha Arns Steiner.

Bibliografia: p. 92-96.

1. Otimização combinatória. 2. Mineração de dados (Computação). 3. Percepção de padrões. I. Universidade Federal do Paraná. II. Steiner, Maria Teresinha Arns. III. Título.

CDD: 519.64

TERMO DE APROVAÇÃO

GENIVAL PAVANELLI

EXTRAÇÃO DE REGRAS DE CLASSIFICAÇÃO DE BASES DE DADOS POR MEIO DE PROCEDIMENTOS META-HEURÍSTICOS BASEADOS EM GRASP

Tese aprovada como requisito parcial para obtenção do grau de doutor no Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof.ª Dr.ª Maria Teresinha Arns Steiner.
(Orientadora) Membro do PPGMNE/UFPR

Prof. Dr. Anderson Roges Teixeira Góes.
Membro do Dep. de Expressão Gráfica da UFPR.

Prof. Dr. Cassius Tadeu Scarpin.
Membro do PPGMNE/UFPR.

Prof.ª Dr.ª Deise Maria Bertholdi Costa.
Membro do PPGMNE/UFPR.

Prof. Dr. Júlio Cesar Nievola.
Membro do PPGEPS/PUCPR.

Curitiba, 28 de maio 2014.

Dedico esse trabalho primeiramente a Deus,
por ser essencial em minha vida, autor de meu destino,
meu guia, socorro presente nas horas de angústia;
à minha esposa Alessandra, pessoa com quem amo partilhar a vida e
à meus filhos, Giovana e Alberto, que iluminam de maneira especial os meus dias.

AGRADECIMENTOS

Com grata satisfação, registro meus agradecimentos a todos que compartilharam o trilhar deste caminho percorrido, contribuindo, de maneira imprescindível para que eu realizasse esta pesquisa, apoiando-me e dando-me forças nos momentos em que mais precisei.

Agradeço, em primeiro lugar, a Deus, por estar comigo em todos os momentos iluminando-me, sendo meu refúgio e fortaleza nos momentos mais difíceis.

Minha gratidão, especialmente, à minha família, pelo apoio para que eu concretizasse esse trabalho. À minha esposa Alessandra, pois ao seu lado aprendi a amar e a sentir-me amado, aprendi a respeitar e ser respeitado e principalmente aprendi que não podemos ter medo de lutar para conquistar. Aos meus filhos Giovana e Alberto, pois a luz e o brilho dos teus sorrisos e olhares me dão motivação para acordar todo dia e enfrentar o que esse mundo tem a nos trazer. Minha amada família, as alegrias de hoje também são suas, pois seus amores, estímulos e carinhos foram armas para essa minha vitória. A vocês minha eterna gratidão.

Agradeço a minha mãe e meu pai, por todos os momentos dedicados a mim, pelas palavras, pelos conselhos, pelo amor, pela honestidade e pelo afeto. Aos meus irmãos, pelos exemplos e pela amizade.

Agradeço a minha sogra, meus cunhados, cunhadas, sobrinhos e sobrinhas, pessoas importantes no conjunto que cerca minha vida.

Agradeço a minha orientadora, Prof. Dra. Maria Teresinha Arns Steiner, pelo entusiasmo, pela sabedoria e principalmente pelo grande incentivo e orientação que me foram concedidos durante essa jornada.

Agradeço aos professores Anderson, Deise e Cassius. Suas contribuições, sugestões e correções muito engrandeceram a minha tese.

Agradeço ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia; seu corpo docente, discente, coordenação e administração; pelos ensinamentos, convívio e apoio.

A todos, muito obrigado.

“Combati o bom combate, completei a corrida, guardei a fé”

2Tm 4,7

RESUMO

O processo de gestão do conhecimento nas mais diversas áreas – seja em indústrias, hospitais, escolas, bancos, dentre outros – exige constante atenção à multiplicidade de decisões a serem tomadas acerca de suas atividades. Para a tomada de decisões, faz-se necessária a utilização de técnicas científicas que lhes garantam a máxima acurácia. O presente trabalho faz o uso de ferramentas matemáticas que cumpram a finalidade de extração de conhecimento de base de dados. O objetivo é a proposição de uma nova meta-heurística, baseada no procedimento *GRASP* (*Greedy Randomized Adaptive Search Procedure*) como ferramenta de *Data Mining* (*DM*), no contexto do processo denominado *Knowledge Discovery in Databases* (*KDD*) para a tarefa de extração de regras de classificação em bases de dados. Assim, a metodologia aqui proposta possui três grandes blocos segundo o processo *KDD*: pré-processamento dos dados, no qual todos os atributos previsores são codificados de maneira a corresponder a uma ou mais coordenadas binárias; aplicação da meta-heurística propriamente dita para extração de regras de classificação; construção do classificador, momento em que as regras extraídas são ordenadas segundo critérios baseados no “fator de suporte” e na “confiança”. A fim de validar esta proposta, a metodologia foi implementada e aplicada a sete bases de dados distintas, com um número variável de instâncias, de atributos e de classes. Os resultados obtidos apresentam elevada precisão preditiva, atingindo, por exemplo, 98% de acurácia para a base de dados zoo, 97% para a base íris e 94% para a base wine. Buscando ratificar os resultados obtidos, foram estabelecidas comparações entre a meta-heurística aqui proposta e os algoritmos BFTree, RepTree e J4.8, todos de árvore de decisão. A partir destas comparações, observa-se que em seis das sete bases analisadas a proposta implementada é superior, em termos de acurácia, aos algoritmos de árvore de decisão utilizados. Desta forma, conclui-se que a meta-heurística proposta atende os pré-requisitos para a tarefa de extração de conhecimento de base de dados.

Palavras-chave: *Greedy Randomized Adaptive Search Procedure*. *Data Mining*. Extração de Regras.

ABSTRACT

The process of knowledge management in several areas – existing in industries, hospitals, schools, banks, among others - requires constant attention to the multiplicity of decisions to be made about their activities. In order to make decisions, it is necessary to use scientific techniques that will ensure their maximum accuracy. This study makes use of mathematical tools that meet the purpose of extracting knowledge from a database. The aim is to propose a new metaheuristic based on GRASP (Greedy Randomized Adaptive Search Procedure) procedure as a tool of Data Mining (DM) within the context of the process called Knowledge Discovery in Databases (KDD) for the task of extracting classification rules in databases. Thus, the methodology proposed herein has three large blocks according to the KDD process: data pre-processing, in which all predictor attributes are encoded to correspond to one or more binary coordinates; application of the metaheuristic itself for extracting classification rules; construction of the classifier, when the extracted rules are ordered in accordance with criteria based on "support factor" and "trust." In order to validate this proposal, the methodology has been implemented and applied to seven different databases, with a variable number of instances, attributes and classes. The results show high predictive accuracy, reaching, for example, 98% accuracy in the zoo database, 97% for the iris base and 94% for the wine base. Seeking to ratify the results, comparisons between the metaheuristic proposed herein and BFTree, RepTree and J4.8 decision tree algorithms were established. Based on these comparisons, it is observed that in six out of seven analyzed bases the implemented proposal is superior, in terms of accuracy, to the used decision tree algorithms. In this way, it is concluded that the meta-heuristic proposed meets the prerequisites for the task of extracting knowledge from a database.

Keywords: Greedy Randomized Adaptive Search Procedure. Data Mining. Rule Extraction.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

CTPS	-	Carteira de Trabalho e Previdência Social
DM	-	*Data Mining*
EGA	-	*Enhanced Genetic Algorithm*
GRASP	-	*Greedy Randomized Adaptive Search Procedure*
GRASP-DM	-	*GRASP* para *Data Mining*
ILS	-	*Iterated Local Search*
KDD	-	*Knowledge Discovery in Databases*
LRC	-	Lista Restrita de Candidatos
MD	-	Mineração de Dados
MO	-	Multi Objetivo
NSGA II	-	*Non-dominated Sorting Genetic Algotithm-II*
PAB	-	Problema de Alocação de Berços
PCV	-	Problema do Caixeiro Viajante
PCVG	-	Problema do Caixeiro Viajante com Grupamentos
PDM	-	Problema da Diversidade Máxima
PEC	-	Problema de Empacotamento de Conjuntos
PML	-	Problema da Mínima Latência
PQA	-	Problema Quadrático de Alocação
PR	-	*Path Relinking*
PS	-	Processo Sumaríssimo
PSET	-	*Parallel machines, Setup times, Earliness e Tardiness*
PST	-	*Parallel machines, Setup times e Tardiness*
RC	-	Reconexão de Caminhos
RCL	-	*Restricted Candidate List*
RO	-	Recurso Ordinário
RR	-	Recurso de Revista
RT	-	Rito de Trabalho
SPEA 2	-	*Strength Pareto Evolutionary Algotithm 2*

SVM	-	*Support Vector Machines*
TPP	-	*Traveling Purchaser Problem*
TRT	-	Tribunal Regional do Trabalho
TS	-	*Tabu Search*
TSP	-	*Traveling Salesman Problem*
TST	-	Tribunal Superior do Trabalho
VNS	-	*Variable Neighborhood Search*
WEKA	-	*Waikato Environment for Knowledge Analysis*
2PNPD	-	*2-Path Network Design Problem*

LISTA DE ABREVIATURAS

Art	-	Artigo
Conf	-	Confiança
ConfMin	-	Confiança Mínima
NIt	-	Número de Iterações
Sup	-	Suporte
SupMin	-	Suporte Mínimo

LISTA DE SÍMBOLOS

$\Rightarrow$	-	Implicação
$\subset$	-	Está contido
$\varnothing$	-	Vazio
$\neq$	-	Diferente
$\cap$	-	Intersecção
α	-	Alfa
c^{max}	-	Custo máximo
c^{min}	-	Custo mínimo
ϵ	-	Pertence
Δ	-	Delta
s^{max}	-	Suporte máximo
s^{min}	-	Suporte mínimo

SUMÁRIO

1 INTRODUÇÃO

Atualmente grande parte das operações e atividades de diversas organizações – sejam das áreas da indústria, comércio ou serviços – é efetivada computacionalmente, fato que gera uma imensa quantidade de dados. À medida que aumentam os bancos de dados oriundos destas transações, cada um dos quais com grande quantidade de dados/registros, aumenta também o interesse e o desafio de extrair destes vastos bancos de dados o conhecimento ali implícito.

Neste contexto, aborda-se neste trabalho o processo de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*; *KDD*), que trata de um processo de descoberta de padrões e tendências por análise de conjuntos de dados. Das etapas que constituem este processo, destaca-se como a mais importante a Mineração de Dados (*Data Mining*; *DM*), que faz uso de algoritmos para extrair conhecimento a partir de grandes volumes de dados, descobrindo relações ocultas, padrões e gerando regras (FAYYAD *et al*., 1996).

Com isso é realizada uma adaptação ao Procedimento de Busca Gulosos Aleatórios e Adaptativos (*Greedy Randomized Adaptive Search Procedure* - *GRASP*) (FEO; RESENDE, 1995; PITSOULIS; RESENDE, 2002; RESENDE; RIBEIRO, 2002; e RESENDE; SILVA, 2013), como ferramenta de *DM*, para executar a tarefa de extração de regras de classificação em bases de dados. Optou-se pelo *GRASP*, por se tratar de uma meta-heurística multi-partida (RESENDE; SILVA, 2013) que aplica repetidamente o método de busca local a partir de soluções construídas por um algoritmo aleatório e guloso. Além disso, trata-se de uma meta-heurística de fácil implementação e que é largamente utilizada em problemas de otimização combinatória.

Atualmente, o *GRASP* possui aplicações nos mais variados campos, dentre os quais se podem citar:

- Problemas de Lógica (FESTA *et al*., 2007, *apud* RESENDE; SILVA, 2013);
- Problemas de Atribuição Quadrática Generalizada (MATEUS *et al*., 2011, *apud* RESENDE; SILVA, 2013);
- Problemas de Alocação (YIN; WANG, 2012; CAPDEVILLE; VIANNA, 2013; ZVIETCOVICH; CARDOSO; MANSO, 2013);
- Elaboração de tabela de horário (ROCHA *et al*., 2012);

- Problema de Roteamento de Veículos (FRANCO JÚNIOR; OLIVEIRA, 2012; OLIVEIRA *et al*., 2012);
- Problema de Ordenação Linear (CHAOVALITWONGSE *et al*., 2011);
- Problema das *P*-Medianas (ZEFERINO; AMORIM; FILHO, 2011);
- Problema de Corte (COELHO *et al*., 2011);
- Reconhecimento de Imagens (HIRSCH; PARDALOS; RESENDE, 2011);
- Problemas de Programação de Tarefas em Máquinas Paralelas (FRANÇA, 2007);
- Problema do Caixeiro Viajante (GONÇALVES; MARTINS; OCHI, 2004);
- Problema de Extração de Regras de Calssificaçãoi (PAVANELLI *et al*., 2014)

1.1 OBJETIVOS DO TRABALHO

Para desenvolver o presente trabalho, foram estabelecidos alguns objetivos que nortearam a pesquisa.

1.1.1 Objetivo Geral

O objetivo principal deste trabalho é construir um classificador a partir de regras obtidas por meio de uma meta-heurística baseada no procedimento *GRASP*.

1.1.2 Objetivos Específicos

A fim de se atingir o objetivo geral proposto, listam-se os seguintes objetivos específicos:

- Aplicar o processo *KDD* como esteio para a construção do classificador a fim de tornar explícitas relações entre atributos preditores e atributo classe;
- Elaborar, implementar e apresentar uma meta-heurística, baseada no procedimento *GRASP*, como método de *DM*, visando extrair regras de classificação em bases de dados;

- Estabelecer um procedimento de ordenação das regras obtidas a fim de que o classificador apresente elevada precisão preditiva;
- Avaliar a qualidade do conhecimento descoberto em termos de precisão preditiva;
- Testar a capacidade de adaptação da metodologia proposta – que envolve a extração de regras e a construção do classificador – aplicando-a em bases de dados distintas.

1.2 JUSTIFICATIVA

Nas mais diversas áreas da indústria, do comércio e do serviço são continuamente geradas imensas quantidades de dados que muitas vezes não são bem “aproveitadas”. Esses dados guardam relações entre si que, se explicitadas, podem auxiliar na tomada de decisões.

Um banco de dados de um hospital, por exemplo, guarda dezenas, centenas e até milhares de dados de pacientes que apresentaram determinados sintomas que induzem a determinado diagnóstico. Já no âmbito das varas do trabalho, têm-se arquivos com grande número de processos, cada qual com uma série de dados. As instituições financeiras apresentam em seus arquivos dados acerca de clientes adimplentes e inadimplentes. Todas essas bases de dados, cada uma com suas especificidades, guardam relações desconhecidas que, se explicitadas, poderão auxiliar os administradores em suas tomadas de decisões.

Assim, justifica-se este trabalho pela sua proposta de elaboração de uma nova metodologia; baseada no procedimento *KDD*, utilizando a meta-heurística *GRASP* como método de *DM*, para a tarefa de classificação. A partir desta metodologia se estabelecem relações entre os dados de uma base de dados, sob forma de regras, que geram conhecimento que podem auxiliar nas tomadas de decisões diante de uma nova situação apresentada.

1.3 LIMITAÇÕES DO TRABALHO

O desenvolvimento dessa tese apresenta algumas limitações inerentes à inovação da meta-heurística proposta:

- As sete bases de dados utilizadas na aplicação da metodologia proposta neste trabalho apresentam um reduzido número de amostras, variando de 101 a 768 padrões.
- O número reduzido de atributos das bases de dados analisadas, que variou de 4 a 16, também é uma das limitações deste trabalho.
- A comparação da metodologia proposta neste trabalho apenas com algoritmos de árvore de decisão também pode ser interpretada com uma de suas limitações.

Apesar das limitações apresentadas acima, as quais surgiram no desenvolvimento desse trabalho, pode-se concluir, a partir dos resultados aqui apresentados, que nenhuma delas foi obstáculo para validar a metodologia proposta e as conclusões que se obtiveram a partir da análise dos mesmos.

1.4 ESTRUTURA DO TRABALHO

Buscando melhor apresentação, clareza, coerência e encadeamento de ideias, este trabalho se divide em cinco capítulos.

Além deste capítulo, apresenta-se, no capítulo 2, a revisão de literatura, momento em que são apresentados conceitos que envolvem as técnicas aplicadas no presente trabalho. Primeiramente, são definidos conceitos de *KDD* com ênfase em *DM*. Dentro das tarefas de *Data Mining*, busca-se priorizar o estudo sobre a extração de regras de classificação – principal objetivo deste trabalho. Neste sentido são discutidas as medidas de desempenho “suporte” e “confiança”, as quais foram utilizadas para avaliar a qualidade das regras obtidas pela heurística proposta. Acerca do *GRASP* são apresentadas inicialmente as duas fases que o compõem: primeiramente, a fase de construção; em seguida, a fase de busca local.

Ainda no capítulo 2 são apresentados vários trabalhos correlatos, ou seja, breves resumos acerca de trabalhos que apresentam aplicações da meta-heurística *GRASP*, principal ferramenta matemática aplicada nesta tese.

No capítulo 3 apresenta-se a descrição da metodologia proposta, ou seja, a construção do classificador a partir de regras obtidas pela meta-heurística baseada no procedimento *GRASP*. Destacam-se os esclarecimentos acerca da proposta em

relação às fases de construção e busca local, bem como a geração da Lista Restrita de Candidatos (LRC) e os métodos de avaliação das regras obtidas.

No capítulo 4 são apresentadas as análises dos resultados da aplicação da heurística proposta por meio da abordagem de sete bases de dados.

Finalmente, no capítulo 5 são apresentadas as conclusões, bem como algumas sugestões para trabalhos futuros.

2 REVISÃO DE LITERATURA

A presente revisão de literatura está dividida em duas grandes partes. A primeira parte trata da abordagem das técnicas envolvidas nesta tese, tais como: Descoberta de Conhecimento em Base de Dados, destacando-se sua definição, as etapas que a constituem e, por fim, as tarefas e métodos desse processo; Procedimento de Busca Gulosos, Aleatórios e Adaptativos, mais precisamente sua definição, bem como as duas fases que o compõem; as medidas de interesse aplicadas na avaliação da metodologia proposta, ou seja, fator de suporte e confiança. A segunda parte apresenta os trabalhos correlatos ao aqui desenvolvido.

2.1 TÉCNICAS APLICADAS

As técnicas utilizadas neste trabalho são alicerçadas no processo *KDD* (*Knowledge Discovery in Databases* ou Descoberta de Conhecimento em Bases de Dados). Na etapa de *DM* (*Data Mining* ou Mineração de Dados), principal etapa do processo *KDD*, é apresentada a proposta principal deste trabalho: uma meta-heurística baseada em *GRASP*, para a tarefa de extração de regras de classificação. Desta forma, cada um destes tópicos é abordado a seguir.

2.1.1 Descoberta de Conhecimento em Base de Dados

O *KDD* é um processo não trivial de descoberta de padrões válidos, novos, úteis e acessíveis, obtidos a partir dos dados armazenados em uma base e que são previamente desconhecidos (FRAWLEY, PIATETSKY-SHAPIRO e MATHEUS, 1991), (FAYYAD *et al.,* 1996) ou, ainda, trata-se da tradução dos dados brutos em informações relevantes (VIANNA *et al.,* 2010). Em outras palavras, trata-se de um processo de extração de informação a partir de dados de uma base de dados que contenha um conhecimento implícito, inicialmente desconhecido, compreensível e potencialmente útil.

Atualmente, a maioria das operações e atividades das instituições privadas, públicas e de economia mista – sejam das áreas da indústria, comércio ou serviço – tem seus registros armazenados eletronicamente, fato que gera grandes bases de dados. As técnicas de *DM* são alternativas eficazes para se descobrir conhecimento

a partir de grandes bancos de dados, dos quais podem ser extraídas relações ocultas que venham a gerar regras para classificar e correlacionar dados que podem ser de grande utilidade nas tomadas de decisões. A obtenção destas relações dá-se de maneira automatizada, o que implica maior rapidez e maior grau de confiança.

Inicialmente, *DM* surgiu como sinônimo de *KDD*. Essa similaridade é errônea, uma vez que a *DM* é apenas uma das etapas do processo *KDD* (GÓES; STEINER, 2012), a qual se relaciona com a análise de dados e o uso de ferramentas computacionais na busca de padrões (características, regras e regularidades) em uma grande base de dados. Segundo Steiner *et al*. (2006), o conhecimento a ser descoberto deve satisfazer a três propriedades: correção (deve ser o mais correto possível); compreensão (deve ser compreensível pelos usuários); atualidade (deve ser interessante, útil e novo). Ainda, segundo Steiner *et al*. (2006), o método aplicado à descoberta do conhecimento deve ser eficiente (acurado), genérico (aplicável a diversos tipos de dados) e flexível (facilmente adaptável).

2.1.1.1 Etapas do processo *KDD*

Segundo Fayyad *et al*. (1996), o processo *KDD* é composto de cinco etapas: seleção dos dados; pré-processamento e limpeza dos dados; formatação ou transformação dos dados; mineração de dados; interpretação e avaliação dos resultados.

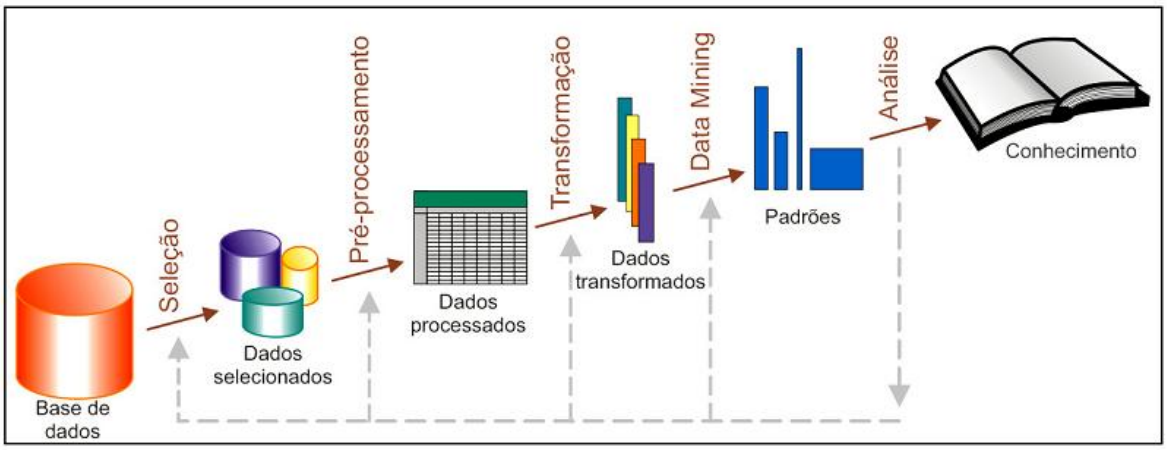

FIGURA 1 – ETAPAS DO PROCESSO *KDD*, ADAPTADA DE FAYYAD *et al*.(1996)
FONTE: Góes e Steiner (2012)

A Figura 1 apresenta a interação entre as cinco etapas do processo *KDD*, segundo Fayyad *et al*. (1996), as quais são especificadas a seguir.

A primeira etapa trata da seleção dos dados (atributos e registros) a serem analisados. Geralmente a seleção é executada por um especialista da área, pois requer conhecimento do assunto e possui papel fundamental no resultado final do processo.

Na segunda etapa, denominada pré-processamento ou limpeza dos dados, eliminam-se os dados redundantes, discrepantes e aqueles que possuem ruídos detectáveis. Nesta etapa podem ser aplicados métodos que levem à redução de dimensionalidade dos dados e/ou melhoria na eficácia dos algoritmos de mineração de dados.

A terceira etapa – transformação dos dados – consiste na formatação adequada e no armazenamento destes dados a fim de aplicá-los no algoritmo da próxima fase.

A quarta etapa é a da Mineração de Dados (*Data Mining*), a fase mais importante do processo (STEINER *et al.*, 2006), na qual são aplicadas as heurísticas ou meta-heurísticas para se extraírem padrões.

A quinta e última etapa trata da interpretação dos resultados, momento em que se busca validar o conhecimento extraído e a eficácia do método aplicado na etapa do *DM* (FAYYAD *et al.*, 1996).

Por se tratar da etapa mais importante do processo *KDD* e, também, por representar o foco deste trabalho, a etapa de mineração de dados é detalhada a seguir.

2.1.1.2 Tarefas e métodos da mineração de dados do *KDD*

Os vários processos desenvolvidos em *DM* têm como objetivo inicial a predição e/ou a descrição. Enquanto a predição usa atributos para predizer os valores futuros de outro atributo, a descrição relaciona o que foi descoberto a partir da base de dados sob o ponto de vista da interpretação humana (FAYYAD *et al*., 1996).

Os objetivos da descrição e da predição são atendidos através de algumas das tarefas principais de *DM* (FAYYAD *et al*., 1996; LAROSE, 2005; GALVÃO; MARIN, 2009), as quais se utilizam de métodos para extrair o conhecimento dos dados. Os métodos de *DM* mais populares são: árvores de decisão, redes neurais

artificiais, *Support Vector Machines* (*SVM*), métodos estatísticos, algoritmos genéticos e, de uma maneira geral, as meta-heurísticas (VALE *et al*., 2008).

A seguir são descritas as seguintes tarefas de *DM*: classificação (foco deste trabalho), regras de associação e agrupamento (*clustering*).

- Classificação:

A classificação, também denominada de aprendizado supervisionado (GOLDSCHMIDT; PASSOS, 2005), certamente é a tarefa mais estudada ao longo do tempo. A tarefa de classificação consiste em alocar um padrão (ou instância) a uma determinada classe dentre outras previamente estabelecidas.

Ao se estabelecerem classes dentro de um conjunto de dados, busca-se corresponder cada uma delas a um conjunto de valores dos atributos previsores, valores esses considerados descritores da classe. Dessa forma, usando os descritores de cada uma das classes que constituem o conjunto de dados é possível construir um classificador que descreve cada instância como pertencente à determinada classe.

Ao se construir um classificador busca-se encontrar as relações entre os atributos previsores e as classes já estabelecidas (WITTEN; FRANK, 2005). A Figura 2 apresenta um classificador cujo objetivo é identificar a relação existente entre os atributos previsores ("*x"* e "*y"*) e as classes ("*a"* e "*b"*).

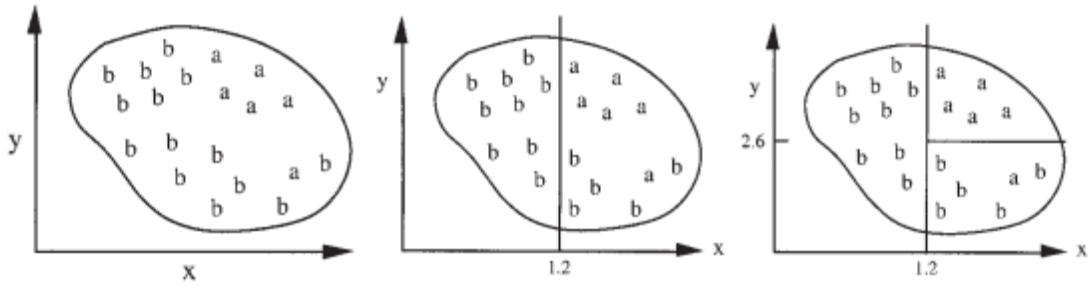

FIGURA 2 - PROCESSO DE CLASSIFICAÇÃO
FONTE: Witten; Frank (2005)

A construção deste classificador é baseada em particionamentos do espaço de dados. A cada estágio, o espaço de dados é dividido em áreas e estas em subáreas, a fim de se obter uma separação das classes. Como se observa na Figura 2, a partir do atributo previsor *x* faz-se o primeiro particionamento do espaço de

dados (x = 1.2). Neste momento, o classificador já estabeleceu uma primeira classificação. Utilizando-se o espaço já particionado pelo atributo x, aplica-se outra divisão do conjunto de dados, desta vez a partir do atributo y (y = 2.6), fase em que o classificador aprimorou a divisão do espaço de dados.

Um classificador de um conjunto de dados pode apresentar dois objetivos: prever um valor alvo e/ou descrever a relação entre os atributos previsores e a classe. Para atingir o segundo objetivo faz-se necessário que o classificador, além de classificar, explicite, de forma compreensível, o conhecimento extraído da base de dados.

A compreensão do conhecimento descoberto dá-se pela descrição da relação entre os atributos e as classes. Essa descrição é geralmente representada na forma de regras “SE” (condições) “ENTÃO” (classe). Essas regras são interpretadas da seguinte maneira: “SE” os valores dos atributos satisfazem as condições da regra, “ENTÃO” a instância pertence à classe prevista pela regra. Voltando a Figura 2, podem-se observar as seguintes regras que descrevem as relações entre os atributos previsores e a classe: SE (x <= 1.2) ENTÃO (Classe b), ou SE (x > 1.2 E y <= 2.6) ENTÃO (Classe b) ou, ainda, SE (x > 1.2 E y > 2.6) ENTÃO (Classe a).

As regras de classificação serão adotadas neste trabalho como forma de representação do conhecimento, em virtude de ser facilmente compreensível para o usuário.

As regras obtidas a partir da tarefa de classificação devem ser genéricas. Para isso é muito importante que se estabeleça a divisão da base de dados em dois grupos: dados de treinamento e dados de teste. Inicialmente, o conjunto de dados de treinamento é disponibilizado e analisado e, a partir deste grupo, se constrói um modelo de classificação. Então, este modelo é aplicado para classificar os dados do conjunto de teste visando validá-lo.

É importante observar que o modelo construído a partir do conjunto de dados de treinamento só será considerado um bom modelo se classificar corretamente uma alta porcentagem das instâncias do conjunto de dados de teste. Concluindo, o modelo deve representar o conhecimento a partir de regras de boa qualidade, ou seja, que possam ser generalizadas para dados que não foram utilizados durante o treinamento.

- Associação:

A tarefa de busca de regras de associação tem por objetivo estabelecer relacionamentos entre itens de uma base de dados (KAZIENKO, 2009). Regras de associação são expressões do tipo $A \Rightarrow B$, que significam: SE (A), ENTÃO (B), em que A e B são conjuntos de itens pertencentes a uma base de dados D, tais que: $A \subset D$, $B \subset D$, $A \neq \emptyset$, $B \neq \emptyset$ e $A \cap B = \emptyset$. Assim, o significado de uma regra de associação ($A \Rightarrow B$) é de que os conjuntos de itens A e B ocorrem juntos em uma mesma transação.

Em uma regra de associação do tipo $A \Rightarrow B$, denota-se por (A) o conjunto de itens no antecedente da regra e por (B) o conjunto de itens no consequente da regra. Dessa forma, de acordo com Goldschmidt e Passos (2005), uma regra de associação indica que o conjunto de itens do antecedente das regras tem propensão a ocorrer juntamente com o conjunto de itens do consequente.

Nas mais diversas áreas, são possíveis aplicações de extração de regras de associação: no problema clássico de cestas de compras (*market basket analysis*), no comércio digital, na navegação em sistemas de informações (WEB), em várias áreas da medicina, em serviços bancários, na detecção de fraudes em cartões de crédito, no gerenciamento de projetos (KAZIENKO, 2009; KARABATAK; INCE, 2009; SÁNCHEZ *et al.,* 2009; GARCIA *et al*., 2008; RIBEIRO *et al*., 2008; METWALLY; AGRAWAL: ABBADI, 2005; AGGELIS, 2004).

Após definidas as tarefas de classificação e de estabelecimento de regras de associação, cabe ressaltar as suas diferenças: a classificação está relacionada à questão da predição, ou seja, analisa o “passado” a fim de inferir acerca do “futuro”; o estabelecimento de regras de associação não envolve predição, mas sim relação.

Da observação da sintaxe das regras, podem-se extrair duas diferenças básicas. Primeiramente as regras de classificação têm apenas um atributo em seu consequente, enquanto regras de associação podem ter mais de um item no seu consequente. A outra diferença reside no fato de que, na classificação, os atributos previsores podem ocorrer apenas no antecedente da regra e o atributo meta pode ocorrer apenas no consequente da regra, ou seja, trata-se de uma regra assimétrica com relação aos atributos a serem minerados. Por outro lado, a tarefa de associação é considerada como simétrica com relação aos itens a serem minerados, já que cada item pode ocorrer tanto no antecedente como no consequente da regra.

- Agrupamento (*Clustering*):

A tarefa de *clustering*, também por vezes chamada de classificação não-supervisionada, tem por objetivo identificar um conjunto finito de classes (*clusters*), a partir de atributos de objetos não classificados previamente. Dessa forma, define-se um *cluster* como um conjunto de objetos agrupados em função de suas similaridades ou proximidades. Os objetos são agrupados buscando-se, por um lado, maximizar as similaridades dentro de um mesmo *cluster* (*intraclusters*) por outro, minimizar as similaridades entre *clusters* diferentes (*interclusters*) (GALVÃO; MARIN, 2009).

A Figura 3 mostra um exemplo do resultado de uma tarefa de *clustering*, na qual três *clusters* foram estabelecidos.

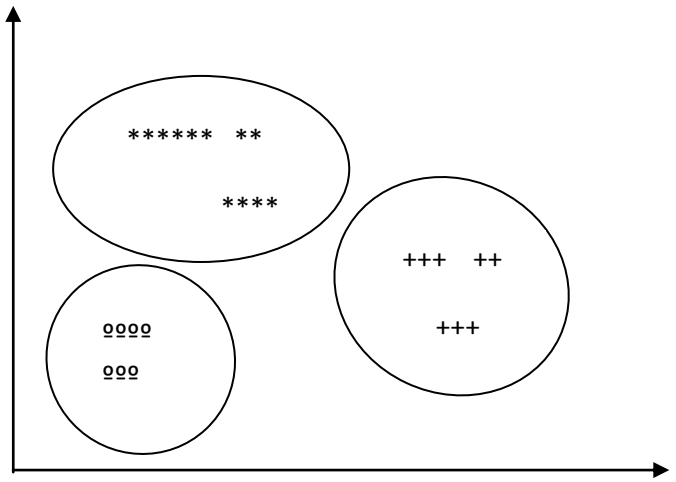

FIGURA 3 – EXEMPLO DE *CLUSTERING* COM TRÊS *CLUSTERS*
FONTE: O Autor (2013)

Quando os *clusters* são estabelecidos, os objetos são atribuídos a seu *cluster* correspondente, e as características comuns dos objetos no *cluster* podem ser reunidas de acordo com os seus principais indicativos para formar a descrição da classe. Um exemplo clássico trata de um conjunto de pacientes que podem ser agrupados de acordo com seus sintomas em várias classes (*clusters*). Assim, cada *cluster* reúne pacientes com sintomas comuns e, dessa forma, poderá ser usado para indicar a qual deles um novo paciente pertencerá. Além disso, ao se utilizar a tarefa de *clustering* para identificar novas classes, esses resultados podem também ser utilizados como pré-processamento para realização da tarefa de classificação (GALVÃO; MARIN, 2009).

2.1.2 Procedimento de Busca Gulosos, Aleatórios e Adaptativos

Nessa seção são apresentados conceitos que envolvem a meta-heurística iterativa de multi-partidas para problemas de otimização combinatória (RESENDE; SILVA, 2013) denominada Procedimento de Busca Gulosos, Aleatórios e Adaptativos (do inglês *Greedy Randomized Adaptive Search Procedure*, *GRASP*). Cada iteração é composta de duas fases: uma de construção e a outra de busca local (FEO; RESENDE, 1995; PITSOULIS; RESENDE, 2002; RESENDE; RIBEIRO, 2002). A fase de construção consiste na elaboração de uma solução factível aleatória, a qual sofrerá um processo de busca local na próxima fase. Esse processo é repetido várias vezes, e a melhor solução dentre todas as iterações é selecionada como resultado final do *GRASP*.

A Figura 4, a seguir, ilustra os principais passos do procedimento *GRASP*.

```
procedure grasp()

1  InputInstance();
2  for GRASP stopping criterion not satisfied
3      ConstructGreedyRandomizedSolution(Solution);
4      LocalSearch(Solution);
5      UpdateSolution(Solution,BestSolutionFound);
6  rof;
7  return(BestSolutionFound)
end grasp;
```

FIGURA 4 – PSEUDOCÓDIGO DO ALGORITMO *GRASP*
FONTE: Feo e Resende (1995)

O pseudocódigo do *GRASP* (Figura 4) apresenta sucintamente os seguintes passos: a linha 1 corresponde aos dados de entrada do problema; as linhas 2 a 6 apresentam o processo iterativo, que termina quando o critério de parada é atingido; na linha 3, tem-se a fase de construção e, na linha 4, a fase de busca local. Se ocorrer uma melhoria na solução, a atualização é apresentada na linha 5. Finalmente, na linha 7, a melhor solução retorna como resultado.

Na fase de construção *GRASP*, a solução é obtida de forma iterativa, ou seja, a cada iteração desta fase, um elemento é acrescido à solução parcial até obter-se a solução completa. Os candidatos a comporem a solução são obtidos a

partir do conjunto de elementos que não comprometem a viabilidade da solução. A avaliação do próximo elemento a compor a solução parcial é feita a partir de uma função de avaliação gulosa.

Os elementos mais bem avaliados por esta função gulosa (aspecto guloso do algoritmo) compõem a Lista Restrita de Candidatos (LRC), do inglês "*restricted candidate list*" (*RCL*), cujo tamanho é definido pelo parâmetro *α*. A partir da LRC, seleciona-se aleatoriamente (aspecto aleatório do algoritmo) o próximo elemento a compor a solução parcial do problema. Na próxima iteração desta fase os elementos que restaram são novamente avaliados, e a LRC é atualizada (aspecto adaptativo do algoritmo), com acréscimo de mais um elemento na solução parcial, até que a solução esteja completa.

A Figura 5, a seguir, ilustra os principais passos da fase de construção do procedimento *GRASP*.

```
procedure ConstructGreedyRandomizedSoluction(Solution)

1  Solution = {};
2  for Solution construction not done
3      MakeRCL(RCL);
4      s = SelectAtRandom(RCL);
5      Solution = SolutionU{s};
6      AdaptGreedyFunction(s);
7  rof;
end ConstructGreedyRandomizedSoluction;
```

FIGURA 5 – PSEUDOCÓDIGO DA FASE DE CONSTRUÇÃO DO *GRASP*
FONTE: Feo e Resende (1995)

O pseudocódigo da fase de construção do *GRASP* (Figura 5) mostra que o processo de construção da solução aleatória gulosa do *GRASP* parte de uma solução inicial que é um conjunto vazio. A LRC é construída com base no parâmetro *α* cujo valor está compreendido no intervalo [0, 1]. A cada iteração desta fase, uma função gulosa avalia os elementos, de maneira que os melhores formam a LRC, cujo tamanho varia de acordo com o parâmetro *α* (aleatório ou guloso). Um elemento (`s`) da lista é escolhido, aleatoriamente, e passa a compor a solução (*Solution*). A função gulosa é adaptada de acordo com a solução parcial. As iterações são

repetidas até que a solução final seja obtida. O final do processo retorna uma solução aceitável ao problema.

No final da fase de construção aleatória e gulosa do *GRASP*, a solução apresentada possivelmente não será um ótimo local. A próxima fase do procedimento *GRASP* tem por objetivo realizar uma busca local nas vizinhanças da solução apresentada na fase anterior a fim de buscar ótimos locais ou, até mesmo, o ótimo global. O algoritmo desta fase também trabalha de modo iterativo, ou seja, a cada iteração busca na vizinhança uma solução melhor do que a apresentada anteriormente.

A Figura 6, a seguir, apresenta o pseudocódigo de busca local partindo da solução (*Solution*) construída na primeira fase do *GRASP* com uma vizinhança *N*(*s*).

```
procedure local(P,N(P),s)
1   for s not locally optimal
2         Find a better solution t ∈ N(s);
3         Let s = t;
4   rof;
5   return(s as local optimal for P)
end local;
```

FIGURA 6 – PSEUDOCÓDIGO DA FASE DE BUSCA LOCAL DO *GRASP*
FONTE: Feo e Resende (1995)

O pseudocódigo da fase de busca local (Figura 6) mostra que o processo de busca local do *GRASP* parte de uma solução inicial (*s*), a qual foi obtida da fase anterior (Fase de Construção). O conjunto *N*(*s*) é composto pelas soluções vizinhas "*t*" da solução apresentada. A cada iteração desta fase, busca-se um ótimo local a partir da comparação dos valores da função objetivo aplicada a (*s*) e a (*t*). No final do processo iterativo desta fase retorna uma solução que é um ótimo local.

A meta-heurística *GRASP* apresenta características atraentes a sua aplicação, tais como a facilidade de implementação e a existência de poucos parâmetros a serem ajustados – basicamente dois: o primeiro relacionado ao critério de parada; o segundo, à cardinalidade da LRC.

O critério de parada do procedimento *GRASP* é normalmente determinado pelo número máximo de iterações, podendo também utilizar-se do tempo de processamento ou, ainda, da obtenção do valor desejado.

O parâmetro *α* está relacionado à cardinalidade da LRC usada na fase de construção da solução do *GRASP*.

A LRC é obtida da seguinte forma: considere um problema de minimização. Seja $D = \{d_1, d_2, \ldots, d_n\}$ um conjunto de elementos a serem acrescidos a uma solução. Define-se $c(d_i)$ o custo da inclusão do elemento d_i à solução. Sejam c^{max} e c^{min} o maior e o menor custo de inclusão, respectivamente. A LRC é composta por elementos d_i pertencentes a D com os menores custos, de maneira que a sua inserção não destrua a viabilidade da solução. A lista fica associada ao parâmetro $\alpha \in [0, 1]$. Os elementos pertencentes à LRC devem incrementar um custo (a solução) menor ou igual a um valor (Δ) pré-definido com base no parâmetro *α*. A equação (1) a seguir define Δ.

$$\Delta = c^{\min} + \alpha(c^{\max} - c^{\min}) \quad (1)$$

Como se pode observar na equação (1), o parâmetro *α* determina quão guloso ou aleatório será o algoritmo de construção (RESENDE; SILVA, 2013). Para *α* igual a “0”, o algoritmo é puramente guloso, enquanto para *α* igual a “1”, o algoritmo é puramente aleatório.

2.1.3 Medidas de interesse: Fator de Suporte e Confiança

As regras de classificação obtidas neste trabalho são geradas a partir de um algoritmo de construção. Para se avaliar se um determinado atributo previsor fará parte ou não da regra em construção, é utilizada a medida de interesse “fator de suporte”. Além dessa medida, outra medida de interesse denominada “confiança” foi utilizada ao final da construção da regra a fim de avaliar a sua qualidade. Essas medidas empregam índices estatísticos para avaliar a força de cada regra. De acordo com Gonçalves (2005), Gonçalves e Plastino (2004) e Gonçalves *et al*. (2004), essas medidas têm por objetivo identificar se as regras são realmente relevantes e úteis.

2.1.3.1 Fator de Suporte

Seja a regra de classificação A ⇒ B, em que A representa os atributos preditores e B o atributo alvo (classe), todos pertencentes a uma base de dados D, tais que A ⊂ D, B ⊂ D, A ≠ Ø, B ≠ Ø e A ∩ B = Ø. Assim, "a regra A ⇒ B tem suporte *s* junto à base de dados D se *s*% das instâncias da base D contém A U B". Portanto, o fator de suporte, ou apenas "suporte", de uma regra representa o percentual de padrões da base de dados que contêm o antecedente (atributos preditores) e o consequente (classe) da regra simultaneamente (SEMAAN; OCHI, 2011; GONÇALVES, 2005).

A Tabela 1 apresenta um conjunto didático de dados que tratam de sintomas da dengue. Cada instância (linha) representa um paciente (padrão), as colunas 2, 3, 4 e 5 representam os sintomas e a coluna 6 representa a classe do paciente, ou seja, se está ou não com dengue.

TABELA 1 – EXEMPLO DE CONJUNTO DE DADOS DE SINTOMAS DA DENGUE

Instância	Febre	Dor de Cabeça	Cansaço	Vômito	Dengue
1	Sim	Sim	Não	Sim	Sim
2	Sim	Sim	Não	Não	Não
3	Não	Não	Sim	Sim	Não
4	Sim	Sim	Sim	Sim	Sim
5	Sim	Sim	Não	Não	Sim
6	Sim	Sim	Sim	Sim	Sim

FONTE: O Autor (2013)

Para o cálculo do suporte da regra SE (*Febre* = *Sim* E *Dor de Cabeça* = *Sim*) ENTÃO (*Dengue* = *Sim*), verifica-se qual o percentual de transações que possuem a união dos itens da regra. Neste caso, os antecedentes verdadeiros *Febre* e *Dor de Cabeça* e o consequente (classe) também verdadeiro *Dengue*. De acordo com a Tabela 1, estes itens estão presentes em quatro das seis transações da base de dados, possuindo, assim, suporte com valor aproximado de 0,67.

Acrescentando o item *Vômito* ao antecedente da regra anterior, tem-se: SE (*Febre* = *Sim* E *Dor de Cabeça* = *Sim* E *Vômito* = *Sim*) ENTÃO (*Dengue* = *Sim*). Esta nova regra terá um suporte menor ou igual ao da regra anterior uma vez que foi

originada a partir do acréscimo de um item ao seu antecedente. Como se pode concluir a partir da Tabela 1, o fator suporte da nova regra é igual a 0,50.

2.1.3.2 Confiança

Seja a regra de classificação A ⇒ B, em que A representa o conjunto dos atributos preditores e B o atributo alvo (classe), itens pertencentes a uma base de dados D, tais que A ⊂ D, B ⊂ D, A ≠ Ø, B ≠ Ø e A ∩ B = Ø. Assim, "a regra A ⇒ B tem confiança *c* junto à base de dados D se *c*% das instâncias da base D que contém A também contém B". Portanto, a confiança de uma regra indica o percentual de padrões que contêm o consequente (classe) dentro do conjunto dos padrões que contêm o antecedente (atributos preditores) (SEMAAN; OCHI, 2011; GONÇALVES, 2005).

A confiança (*Conf*) da regra é calculada com base nos suportes (*Sup*) dos antecedentes e da classe, conforme equação (2).

$$Conf\,(A \Rightarrow B) = \frac{Sup(A \cup B)}{Sup(A)} \tag{2}$$

Assim, retornando aos dados da Tabela 1 para se calcular a confiança da regra SE (*Febre* = *Sim* E *Dor de Cabeça* = *Sim*) ENTÃO (*Dengue* = *Sim*), verifica-se qual o percentual de instâncias (pacientes) – dentre os que apresentam o antecedente "*Febre* = *Sim* E *Dor de Cabeça* = *Sim*" – que apresenta o consequente "*Dengue* = *Sim*". Conforme apresentado na Tabela 1, o antecedente está contido em cinco instâncias e, dentre elas, o consequente se encontra em quatro. Desta forma, a confiança desta regra é de 0,80.

Conforme proposto para o cálculo do fator suporte, acrescentando-se o item *Vômito* ao antecedente da regra anterior, tem-se: SE (*Febre* = *Sim* E *Dor de Cabeça* = *Sim* E *Vômito* = *Sim*) ENTÃO (*Dengue* = *Sim*). Comparando-se com os cálculos anteriores, esse acréscimo infere um suporte (a essa regra) menor do que o da regra que a originou, neste caso 0,50; observa-se, porém, que a confiança aumentou de 0,80 para 1,0.

Diante do exposto, buscando atender o requisito de abrangência de uma regra (suporte) e ao mesmo tempo da sua confiabilidade, considera-se uma boa

maneira de se avaliar a qualidade das regras obtidas o modelo suporte-confiança, proposto por (SEMAAN; OCHI, 2011). Este modelo consiste em obter todas as regras que possuam fator de suporte e confiança que satisfaçam aos valores mínimos submetidos como parâmetros.

A avaliação das regras adotada neste trabalho seguiu o modelo suporte-confiança, ou seja, primeiramente são determinadas as regras que satisfaçam o suporte mínimo e, em seguida, a partir deste conjunto de regras, extraem-se as regras que atendam a confiança mínima.

Aplicando o modelo proposto no exemplo didático apresentado na Tabela 1 e estipulando como parâmetros o suporte mínimo igual a 0,50 e a confiança mínima igual a 0,80, podem-se obter várias regras de classificação. A Tabela 2 apresenta algumas destas regras.

TABELA 2 – SUPORTE E CONFIANÇA DE REGRAS DE CLASSIFICAÇÃO

Regra	Antecedente (A)	Classe(B)	Sup (A⇒B)	Conf (A⇒B)
1	Febre E Dor	Dengue	0,67	0,8
2	Febre E Vômito	Dengue	0,50	1,0
3	Febre E Dor E Vômito	Dengue	0,50	1,0
4	Dor E Vômito	Dengue	0,50	1,0
5	Febre E Dor E Cansaço E Vômito	Dengue	0,33	1,0

A Tabela 2 apresenta algumas regras obtidas a partir da Tabela 1 e selecionadas segundo o modelo proposto, estabelecendo-se como parâmetros o suporte mínimo igual a 0,50 e a confiança mínima igual a 0,80. Ao se estabelecer, por exemplo, a regra SE (*Febre* E *Dor de Cabeça* E *Cansaço* E *Vômito*) ENTÃO (*Dengue*), calculam-se as medidas de interesse, obtendo-se fator de suporte igual a 0,33 e confiança igual a 1,0. A partir destes resultados, verifica-se que essa regra não atende ao modelo proposto uma vez que o valor do suporte é inferior ao mínimo pré-estabelecido. Assim, esta regra será descartada.

2.2 TRABALHOS CORRELATOS

A apresentação dos trabalhos correlatos é estabelecida em três tópicos. Primeiramente são apresentados trabalhos com diversas aplicações da meta-heurística *GRASP*, que é a base para o desenvolvimento desta tese. A segunda parte relaciona a meta-heurística *GRASP* com algumas tarefas de *Data Mining*, aproximando-se ainda mais a literatura científica atual do trabalho aqui desenvolvido. O terceiro e último tópico estabelece uma comparação dos trabalhos abordados no tópico anterior com a metodologia proposta neste trabalho.

2.2.1 Aplicações da Meta-heurística *GRASP*

Range, Abreu e Boaventura (2000) estabelece uma proposta, a partir da meta-heurística *GRASP* para o Problema Quadrático de Alocação (PQA), que busca descartar soluções iniciais supostamente ruins com base na normalização de custos calculadas num intervalo entre limites de solução. Trata-se de uma modificação na heurística com uma proposta para aceitar ou não a solução inicial gerada na fase de construção, evitando uma busca que, eventualmente, exigiria muito esforço computacional. Essa meta-heurística foi denominada *GRASP* Restrito. Para comparação dos resultados, foram implementados o *GRASP* e o *GRASP* Restrito. Diversos testes foram realizados, variando-se os limites de aceitação. O *GRASP* Restrito permitiu economia do tempo computacional gasto sem prejuízo à qualidade das soluções alcançadas, ou seja, a melhor solução viável alcançada foi a mesma que o *GRASP* atingiu. A proximidade com a melhor solução viável conhecida foi em média 99%.

Em seu trabalho, Silva, Drummond e Ochi (2000) apresentam algoritmos meta-heurísticos baseados em *GRASP* e *VNS* (*Variable Neighborhood Search*) para solução da generalização do Problema do Caixeiro Viajante (PCV ou *Traveling Salesman Problem*, *TSP*), em que somente um subconjunto de cidades é visitado, o *Traveling Purchaser Problem* (*TPP*). Foram utilizados seis algoritmos de construção e oito algoritmos de busca local para aplicação nas meta-heurísticas *GRASP* e *VNS*. Para o *GRASP* foram utilizados os algoritmos ADDGENI, DROPGENI, RandomADD e RandomDROP, para a geração da solução inicial. Para a busca local, foram

aplicados os algoritmos ADDSearch, DROPSearch, ADDDROPSearch, DROPADDSearch, SwapSearch e Híbrido, totalizando 24 algoritmos do tipo *GRASP*. Para o *VNS* foram utilizados, na geração da solução inicial, além daqueles aplicados ao *GRASP*, também os algoritmos ADD e DROP. Na fase de busca local foram aplicados os mesmos algoritmos de busca do *GRASP*, perfazendo um total de 36 algoritmos do tipo *VNS*. A fim de avaliar os algoritmos *GRASP* e *VNS* propostos, compararam-se seus resultados com os da meta-heurística *Tabu Search* (TS). Verificou-se que, em índices de erro, o algoritmo GRASPVNS1 – que aplica ADDGENI na solução inicial e Híbrido na Busca Local – foi superior aos dos demais. Em relação ao tempo de processamento, as médias do TS foram melhores.

Silva, Drummond e Ochi (2006) propõem combinações de métodos heurísticos de construção e de busca local que são usados como base em diferentes versões do algoritmo *GRASP*, a fim de resolver o Problema de Diversidade Máxima (PDM). A primeira heurística proposta é denominada Heurística Aleatorizada das *K* maiores Distâncias (HA-KMD) que constrói uma solução inicial de forma iterativa em que um elemento é selecionado para ser inserido na solução parcial de forma aleatória, partindo de uma LRC (Lista Restrita de Candidatos) que contém os *K* melhores candidatos. Outra heurística denomina-se Heurística Aleatorizada das *K* Maiores Distâncias 2 (HA-KMD2), que se diferencia da primeira pelo fato de a LRC neste caso ser inteiramente recalculada após a inclusão de cada elemento na solução corrente. A próxima heurística proposta, Heurística Aleatorizada da Inserção mais Distante (HA-IMD), utiliza o conceito clássico da inserção mais distante. A avaliação do desempenho das heurísticas desenvolvidas para o PDM foi realizada a partir das combinações de métodos de construção com dois métodos de busca local. Os resultados mostram que os algoritmos propostos sempre alcançam uma solução ótima quando esta é conhecida e, nos outros casos, apresentam um desempenho superior ao das melhores heurísticas *GRASP* da literatura.

França (2007) tem por objetivo minimizar a soma ponderada de custos de atraso de um conjunto de tarefas (*PST - Parallel machines, Setup times e Tardiness*) a serem processadas em um conjunto de máquinas idênticas e continuamente disponíveis, bem como minimizar os custos de avanço e atraso de um conjunto de tarefas (*PSET- Parallel machines, Setup times, Earliness e Tardiness*), por meio de

GRASP e Busca Tabu, aplicando duas variantes. A primeira variante é em termos de movimentos de troca e de inserção; a segunda, em termos de movimentos *cross* e *Or-Opt*, ambas exploradas alternadamente. Ao final, os resultados de cada variante são comparados buscando-se avaliar o desempenho entre os tipos de movimentos. O aperfeiçoamento do desempenho do *GRASP* é feito a partir de uma combinação entre a função gulosa usada na fase construtiva e uma função de intensificação. Outra abordagem conecta a solução encontrada ao final de uma iteração do *GRASP* a uma solução do conjunto de elite, em um procedimento denominado reconexão de caminhos. Tal procedimento também é utilizado como pós-otimização, ligando as soluções de elite entre si. Na comparação entre a melhor implementação *GRASP* com memória e pós-otimização via reconexões de caminhos e quaisquer das implementações de Busca Tabu com estratégias de longo prazo, foi observado um melhor desempenho da Busca Tabu, tanto para o problema "*Parallel machines, Setup times e Tardiness"* quanto para o problema "*Parallel machines, Setup times, Earliness e Tardiness"*.

Em seu artigo, Mestria, Ochi e Martins (2009) apresentam, além do *GRASP* clássico, outras três versões incorporando memória adaptativa a esta meta-heurística para resolução do Problema do Caixeiro Viajante com Grupamentos (PCVG). A primeira versão utiliza a estrutura do *GRASP* clássico com o método de inserção do mais próximo e na fase de busca local o método 2-*optimal*. A segunda versão apresentada acrescenta à versão anterior a reconexão de caminhos (RC; ou *Path Relinling*) entre todos os pares de solução do conjunto de elite. A terceira versão incorpora à primeira uma RC a cada iteração *GRASP*, ou seja, ao final da busca local aplica-se a RC à solução encontrada com uma das soluções do conjunto de elite. A quarta hibridização apresentada acrescenta à terceira versão uma RC, ao final do *GRASP*, entre todos os pares de solução do conjunto de elite. O desempenho das heurísticas foi avaliado a partir de seis instâncias, comparando-se os resultados obtidos pelas versões apresentadas com a solução ótima (quando possível) ou a melhor solução conhecida. Cada heurística apresentada foi executada 10 vezes para cada instância. Ao final dos testes, observa-se que a quarta versão híbrida é a que apresentou melhor resultado.

Lopes, Schulz e Mauri (2011) apresentam um algoritmo híbrido para a meta-heurística *GRASP* de forma integrada com o método *Path Relinking* (*PR*) para a solução do Problema de Alocação de Berços (PAB). O PAB tem como objetivo minimizar o tempo de serviço de cada navio no porto, que é dado pela soma do tempo de espera com o tempo de atendimento e, para isso, deve selecionar os navios que chegam ao porto e atribuí-los a berços ao longo do cais. Uma solução para o PAB é representada por uma matriz, na qual cada linha representa um berço e as colunas a sequência de atendimento dos navios. Para obtenção desta matriz, cada iteração *GRASP* gera uma solução por meio de um algoritmo de construção na qual é aplicada uma busca local e, a cada intervalo de iterações, é aplicado o *PR* que explora as trajetórias que conectam a solução corrente com a melhor solução encontrada. A avaliação do desempenho do método proposto foi estabelecida e a partir de sua aplicação a 30 instâncias geradas aleatoriamente e distintas, cada uma delas com 60 navios e 13 berços. A cada instância foram aplicados 10 testes aleatórios. As melhores soluções obtidas pelo GRASP+PR foram comparadas com quatro melhores soluções apresentadas em trabalhos recentes; em dois casos foi obtida uma solução melhor e nos outros dois a solução foi similar.

No artigo apresentado por Silva, Subramanian e Ochi (2011), é tratado o Problema da Mínima Latência (PML) a partir de um algoritmo baseado nas meta-heurísticas *GRASP* e *Iterated Local Search* (*ILS*), utilizando-se como método de busca local o *Variable Neighborhood Descent* (*VND*) com ordem aleatória de vizinhanças (*RVND*). A solução inicial viável para o PML é gerada por um procedimento guloso construtivo, baseado na fase de construção do *GRASP*. A fase de busca local é realizada por um método baseado no *VND*, com seleção aleatória da ordem em que as estruturas de vizinhança serão executadas. Cinco estruturas de vizinhanças foram aplicadas: *Swap*, permutação entre dois clientes; 2-*opt*, dois arcos não adjacentes são removidos e outros dois são inseridos; Reinserção, troca-se a posição de um único cliente; *Or-opt* 2, dois clientes adjacentes são removidos e inseridos em outra posição do percurso e *Or-opt* 3, três clientes adjacentes são removidos e inseridos em outra posição do percurso. O algoritmo foi testado em dois conjuntos de instâncias: o primeiro, composto de 22 problemas-teste, com número de clientes variando entre 42 e 107; o segundo, de 10 problemas, variando de 70 a

532 clientes. O método proposto apresentou melhora na solução de 11 problemas e chegou à solução conhecida nas outras 21 instâncias.

Chaovalitwongse *et al.* (2011) propõem um novo algoritmo baseado em *GRASP* para resolver o Problema de Ordenação Linear. Dado um grafo com *n* vértices, *m* arcos e os custos nos arcos, o problema de ordenação linear consiste em encontrar uma permutação π dos vértices, de forma a minimizar o custo dos arcos reversos. Trata-se de um problema de otimização combinatória *NP-hard*, cujo estudo se justifica devido à ampla gama de aplicações deste tipo de problema em economia, arqueologia, ciências sociais, programação e biologia. O novo algoritmo proposto integra o *GRASP* com um procedimento de *Path Relinking*, e foi aplicado a 49 conjuntos de dados reais (instâncias LOLIB) extraídos do repositório Reinelt, que contam com número de objetos variando de 44 até 60. Foram aplicados também a outros 30 casos gerados aleatoriamente e propostos por Mitchell (1997), cujo número de objetos é ainda maior, variando de 100 até 250. Os valores ótimos para todas as instâncias são conhecidos. Para cada instância, o algoritmo foi aplicado cinco vezes com número máximo de iterações *GRASP* igual a 5.000. Foram analisados os resultados para 200 e também para 5.000 iterações. Os autores concluem apresentando os resultados como um desempenho notavelmente robusto, visto que foram obtidas as soluções ótimas para a maioria dos casos LOLIB, e, para as instâncias Mitchell, a diferença entre as soluções encontradas e a solução ótima é em média 0,001%. Quanto ao tempo de processamento para as 200 iterações, as médias foram de 0,14 segundo para as instâncias LOLIB e de 14,3 segundos para as instâncias Mitchell.

Yin e Wang (2012) apresentam um algoritmo GRASP-VNS a fim de otimizar a colocação de turbinas eólicas em parques eólicos, estabelecendo uma relação entre o custo de colocação e a energia produzida por várias turbinas sob várias velocidades e direções de vento. Foram analisados quatro cenários: vento uniforme com direção única, vento uniforme com direção variável, vento não uniforme com direção variável e vento não uniforme de direção variável e condições de solo. No primeiro caso (vento uniforme com direção única), as propostas GRASP-VNS obtiveram a solução global ótima, monstrando-se tão eficazes quanto os métodos concorrentes (Mosetti, Grady, e *EGA – Enhanced Genetic Algorithm*) em identificar o

ótimo global. Também no segundo cenário, que envolve vento com direção variável (foram consideradas 36 direções, de 0º a 360º com incrementos de 10º) e velocidade uniforme, a abordagem GRASP-VNS proposta obtém o melhor valor, indicando que tal método é o mais eficaz entre todos os concorrentes. No terceiro cenário, que descreve a situação de vento com direção e velocidades variáveis (8, 12 e 17 m/s), a abordagem GRASP-VNS proposta supera todos os métodos de comparação, indicando que o algoritmo GRASP-VNS é robusto contra as condições variáveis ambientais observados em parques eólicos. No quarto e último cenário, as condições de solo, como umidade e rochosidade, são adicionadas ao cenário anterior, aproximando-o ainda mais de problemas reais. Neste caso, observa-se que os resultados numéricos entre os métodos *EGA* e GRASP-VNS são semelhantes e os custos obtidos pelos dois métodos são os mesmos, já que ambos os métodos propõem a instalação do mesmo número de turbinas eólicas.

Festa e Resende (2013) apresentam hibridizações de duas meta-heurísticas: *GRASP* e *Path-Relinking*. O objetivo de tais procedimentos é compensar as lacunas de uma meta-heurística com as características especiais da outra. São apresentadas quatro hibridizações: na primeira delas, cada solução *GRASP* é aplicada à reconexão de caminhos com uma solução aleatória do conjunto de elite, dando origem a uma nova solução. Se esta for melhor do que a melhor das soluções do conjunto de elite, substitui a pior solução do conjunto; a segunda técnica apresenta a hibridização *GRASP* com um algoritmo evolutivo de reconexão de caminhos, e difere da anterior pelo fato de aplicar, de acordo com algum critério (normalmente número de iterações), a religação de caminhos dentro do conjunto de elite; a terceira técnica combina uma heurística lagrangeana gulosa a um *GRASP* com reconexão por caminhos; por fim, a quarta técnica apresenta o *GRASP* paralelo com reconexão de caminhos. A meta-heurística *GRASP* não faz uso de qualquer tipo de memória. Ao se aplicar a hibridização com reconexão de caminhos se introduzem estruturas de memória, fato que gera um trabalho extra a cada iteração (manutenção do conjunto de elite e a religação de caminho). A partir dos testes realizados pelos autores, porém, foi verificado que, em se mantendo fixo o tempo de processamento, a probabilidade de o algoritmo *GRASP* com reconexão de caminhos encontrar a solução desejada é maior do que a do o *GRASP* puro. Por outro lado, buscando-se uma solução, a probabilidade de o algoritmo híbrido encontrá-la mais

rapidamente é maior do que a do *GRASP*. Os autores concluem afirmando que "Hibridização com reconexão de caminhos é agora a abordagem padrão para a implementação *GRASP*".

2.2.2 *GRASP* como Método de *Data Mining*

Ribeiro (2005) propõe uma versão híbrida da meta-heurística *GRASP* com técnicas de *DM*, a qual denomina de GRASP-MD, que pode ser considerada como uma meta-heurística encadeada (*relay*) e de alto nível (*high level*), visando introduzir uma memória ao *GRASP*. A proposta da técnica GRASP-MD é inicialmente armazenar um conjunto de soluções sub-ótimas, obtidas a partir de iterações *GRASP*. A partir deste conjunto de soluções, utiliza-se o processo de MD a fim de extrair padrões que ocorrem com frequência. Por fim, estes padrões são utilizados na construção de soluções para as próximas iterações. Para a seleção dos padrões a serem utilizados na fase híbrida do GRASP-MD, foram utilizadas quatro estratégias. A primeira versão, denominada Estratégia do Maior Padrão, consiste na MD do conjunto de elite, que extrai padrões com suporte igual ou superior a um valor pré-estabelecido. Deste conjunto é selecionado o padrão que possui o maior número de itens, o qual será utilizado em todas as iterações da fase híbrida do GRASP-MD. A Estratégia do Maior Padrão para cada Suporte difere da anterior pelo fato de utilizar-se de um conjunto de padrões para a fase híbrida. Tal conjunto é composto pelos padrões com maior número de itens que apresentam suporte maior ou igual ao pré-estabelecido. Assim como a versão anterior, a Estratégia dos *n* Maiores Padrões utiliza-se de um conjunto de padrões para aplicar na fase híbrida. Neste caso, o conjunto é constituído pelos *n* maiores padrões com suporte superior ao valor pré-estabelecido. A quarta estratégia apresentada, denominada *n* Maiores Padrões Maximais, difere da anterior pelo fato de que, durante a geração do conjunto de elite, cada iteração é feita com um valor diferente para o suporte mínimo. Com o objetivo de avaliar se a introdução de MD melhora o desempenho da meta-heurística *GRASP*, foram comparadas as estratégias aqui apresentadas com o *GRASP* puro em 14 instâncias do Problema de Empacotamento de Conjuntos (PEC). A estratégia híbrida apresentou melhores resultados para 11 delas, enquanto o *GRASP* puro apresentou melhor resultado em apenas uma, tendo havido empate em duas das instâncias.

Ribeiro, Plastino e Martins (2006) propõem a incorporação de um processo de mineração de dados à meta-heurística *GRASP.* O objetivo foi aplicar o processo de mineração de dados em um conjunto de soluções sub-ótimas, obtido a partir um determinado número de iterações *GRASP*. A estratégia proposta para hibridizar a meta-heurística *GRASP* com o processo de mineração de dados é composta por duas fases. Primeiramente executam-se, por um determinado tempo de CPU, iterações *GRASP* a fim de obter o conjunto de elite. Deste conjunto se extrai um conjunto padrões que contém os itens mais frequentes. A segunda fase consiste em aplicar o *GRASP* novamente (pelo mesmo intervalo de tempo da primeira fase); porém, cada solução construída irá conter os elementos dos padrões que apresentam os itens mais frequentes, obtidos na fase anterior. Nos testes realizados para Problema de Empacotamento de Conjuntos, tanto na fase de geração do conjunto de elite quanto na fase híbrida, o *GRASP* foi executado por 10.800 s ou 260 iterações. Os resultados obtidos para atingir um valor-alvo mostraram que a estratégia de mineração de dados acelera o processo de se encontrar uma boa solução pelo procedimento *GRASP*.

Fonseca *et al.* (2008) apresentam uma versão híbrida da meta-heurística *GRASP*. Tal versão incorpora técnicas de MD denominada DM-GRASP, aplicada ao problema de replicação de servidores para transmissão *multicast* confiável. Visando investigar quantas vezes e em que momentos deve ser executado o processo de mineração, foram propostas três variações do DM-GRASP. A primeira variação apresenta duas versões: DM 20, em que 20% das iterações *GRASP* são realizadas para, em seguida, aplicar-se o processo de DM; DM 80, momento em que se realizam 80% das iterações *GRASP* para posterior aplicação do processo de *DM*. A segunda variação, denominada DM-3X, executa o algoritmo de mineração três vezes: uma após 20% das iterações e as outras com 50% e 80% das iterações. A terceira variação apresenta as versões DM-D5, DM-D10 e DM-D20; nestas não há momento pré-definido para se executar o algoritmo de mineração. Na versão DM-D5, o algoritmo de mineração é aplicado quando o conjunto de elite não sofre alteração por um número de iterações igual a 5% do total, enquanto para as versões DM-D10 e DM-D20 esse número é igual a 10% e 20%, respectivamente. A fim de se avaliarem os algoritmos, estes foram aplicados a cinco cenários de transmissão

multicast. A versão DM-D5 apresentou melhores resultados e menores tempos de execução.

Fonseca *et al*. (2009) apresentam uma versão híbrida e adaptativa da meta-heurística *GRASP*, denominada MDM-GRASP (*Multi Data Mining GRASP*). Trata-se de uma versão adaptativa do DM-GRASP (*Data Mining GRASP*). No DM-GRASP a mineração de dados é executada apenas uma vez; já no MDM-GRASP o processo de mineração de dados é executado toda vez que o conjunto de elite se torna estável, fato que permite a extração de padrões mais refinados. A validação da versão MDM-GRASP foi estabelecida a partir de dois problemas. O primeiro trata da replicação de servidores para transmissão de *multicast* confiável, aplicado a cinco cenários de transmissão com número de nós variando de 200 a 2000. O segundo foi o problema das *p*-medianas, aplicado a dois grupos com 20 instâncias cada, com *p* variando de 10 a 100, com incremento de cinco. Os resultados obtidos nos dois tipos de problemas apontaram melhores soluções, com um tempo computacional menor para o MDM-GRASP em comparação ao do MD-GRASP e ao do próprio *GRASP*.

Reynolds e Iglesia (2009) propõem a adaptação de algoritmos multi-objetivo (MO) para a tarefa de classificação parcial, bem como introduzem um novo algoritmo MO para esta tarefa baseado na meta-heurística *GRASP*. A técnica *GRASP* faz uso de algumas características do problema de *Data Mining*, buscando uma construção eficaz do conjunto de soluções iniciais, o qual servirá como ponto de partida para a fase de busca local do *GRASP*. Tal algoritmo utiliza-se dos conceitos de dominância e otimalidade de Pareto. Buscando validar o algoritmo proposto, os referidos autores compararam-no com os resultados de diversas técnicas de solução de problemas MO. O desempenho obtido foi pouco superior ao NSGA II (*Non-dominated Sorting Genetic Algorithm*-II) e SPEA2 (*Strength Pareto Evolutionary Algorithm* 2) para pequenos conjuntos de dados. Ao serem aplicadas tais ferramentas em conjuntos de dados maiores, os autores esperam uma maior diferenciação entre as técnicas.

Barbalho *et al*. (2011) também apresentam a hibridização da meta-heurística *GRASP* que incorpora o processo de religação de caminhos (*Path Relinking*) e um módulo de mineração de dados. A aplicação proposta neste trabalho é para o problema 2PNPD (2-*path network design problem*). A contribuição deste trabalho é

mostrar que não só a meta-heurística *GRASP* tradicional, mas também a hibridização do *GRASP* com a heurística *path-relinking* podem se beneficiar da incorporação de um processo de mineração de dados para se extraírem padrões de soluções sub-ótimas a fim de orientar, de forma mais eficiente, a busca de melhores soluções. Foram aplicadas três versões híbridas do *GRASP*. Na primeira delas, denominada GRASP-PR, a religação de caminhos é usada após cada iteração *GRASP*, ligando-se a solução obtida da busca local com uma solução do conjunto de elite. A segunda versão, denominada DM-GRASP-PR, é composta de duas fases: (1) execução de *n* iterações *GRASP* para obtenção do conjunto de elite; (2) denominada híbrida, faz-se nova execução de *n* iterações *GRASP*, porém com a construção guiada por um padrão já extraido do conjunto de elite. A terceira versão apresentada, denominada MDM-GRASP-PR, difere da anterior pelo fato de o processo de mineração de dados não ser executado apenas uma vez, mas sempre que o conjunto de elite se torna estável. Os resultados experimentais mostraram que a primeira versão da estratégia híbrida proposta, chamada DM-GRASP-PR, foi capaz de obter as melhores soluções em menos tempo computacional do que o obtido pelo *GRASP* original com religação de caminho. O MDM-GRASP-PR obteve resultados ainda melhores do que o DM-GRASP-PR.

Em seu trabalho Plastino *et al.* (2011) apresentam uma versão híbrida da meta-heurística *GRASP*, que incorpora técnicas de *Data Mining* para a solução do problema de *p*-medianas, a qual denominaram DM-GRASP (*Data Mining GRASP*). A proposta desta hibridização baseia-se na hipótese de que os padrões extraídos de um conjunto de soluções sub-ótimas podem guiar a busca por soluções melhores. O algoritmo DM-GRASP é composto por duas fases. Na primeira, chamada fase de geração do conjunto de elite, o *GRASP* é executado *n* vezes a fim de obter um conjunto de soluções diferentes. As melhores soluções deste conjunto compõem o conjunto de elite. Neste momento é aplicado o processo de mineração de dados, a fim de se extrairem padrões (conjuntos de elementos) que aparecem com frequência em soluções do conjunto de elite. Na segunda fase, são executadas outras *n* iterações adaptadas, nas quais os padrões são utilizados para guiar a construção das soluções. Buscando melhorar ainda mais as soluções, propõem outra versão, na qual a mineração de dados é realizada não apenas uma vez como no DM-GRASP, mas sempre que o conjunto de elite se torna estável, ou seja, quando não ocorre

nenhuma mudança no conjunto de elite ao longo de um número pré-definido de iterações. Essa versão foi denominada MDM-GRASP (*Multi Data Mining GRASP*). O *GRASP* puro, bem como as versões propostas, foi aplicado a 80 casos de problemas das *p*-medianas com o objetivo de avaliar seus rendimentos. Os experimentos realizados mostraram que o DM-GRASP obteve resultados melhores do que os obtidos pela aplicação do *GRASP* puro e que o MDM-GRASP obteve resultados ainda melhores do que os do DM-GRASP, não só em termos de qualidade, mas também em relação ao tempo computacional.

Semaan e Ochi (2011) apresentam um novo algoritmo heurístico baseado na meta-heurística *GRASP* para a extração de regras de associação. Com o objetivo identificar quais regras são realmente relevantes e úteis, são calculadas medidas de interesse para as regras de associação. Essas medidas empregam índices estatísticos para avaliar a força de cada regra. São elas: o fator de suporte, que representa o percentual de transações da base de dados que contêm os itens do conjunto; a confiança da regra $A \Rightarrow B$ (A implica B), que é um valor que indica, dentre as transações que contêm os itens de *A*, o percentual de transações que também contêm os itens de *B*; o *lift*, que avalia as dependências entre o conjunto de itens antecedentes e consequentes das regras. Na primeira fase da heurística proposta foi considerada a formação de *itemsets* de tamanho *k*, submetido como parâmetro, e não a construção de soluções (regras de associação). A segunda fase – momento em que ocorre efetivamente a extração das regras de associação – atua na construção da solução e refinamento dos *itemsets* construídos na fase anterior. Com o objetivo de diversificar a formação de *itemsets* foram considerados quatro critérios relacionados ao suporte dos itens, quais sejam: (1) Mais Frequentes: selecionam 75% dos *K* itens de maneira aleatória entre os 20% mais frequentes e os demais 25% de fora dessa faixa. (2) Menos Frequentes: selecionam 75% dos *K* itens de maneira aleatória entre os 20% menos frequentes e os demais 25% de fora dessa faixa. (3) Misto: selecionam 50% dos *K* itens de maneira aleatória entre os 50% menos frequentes e os demais de fora dessa faixa. (4) Totalmente aleatório: selecionam quaisquer *K* itens. Os resultados obtidos mostraram que a utilização do algoritmo proposto é uma alternativa interessante para a obtenção de regras de associação de qualidade, ainda que seus *itemsets* possuam baixo(s) suporte e/ou confiança.

2.2.3 Similaridades entre os trabalhos correlatos e a metodologia proposta

Nesta seção, faz-se uma comparação entre os trabalhos correlatos analisados na seção anterior com a proposta desenvolvida nesta tese. A Tabela 3 a seguir apresenta os dados relevantes para esta análise de similaridade.

TABELA 3 – TRABALHOS CORRELATOS *VERSUS* META-HEURÍSTICA PROPOSTA

Autor(es)/ ano	Objetivo	Proposta entre *DM* e *GRASP*	Problema resolvido
1. Ribeiro (2005)	Introduzir memória à meta-heurística *GRASP*	1. Armazenar um conjunto de soluções sub-ótimas obtidas nas iterações *GRASP*. 2. Utilizar o processo de MD a fim de extrair padrões que ocorrem com frequência. 3. Construir soluções *GRASP* a partir dos padrões extraídos na MD.	Problema de Empacotamento de Conjunto
2. Ribeiro Plastino e Martins (2006)	Introduzir memória à meta-heurística *GRASP*	1. Executar, por um determinado tempo de CPU, iterações *GRASP* a fim de obter o conjunto de elite. 2. Utilizar o processo de MD a fim de extrair um conjunto de itens mais frequentes no conjunto de elite. 3. Aplicar o *GRASP* novamente a fim de construir soluções que contenham os itens mais frequentes, obtidos na fase anterior.	Problema de Empacotamento de Conjunto
3. Fonseca *et al.* (2008)	Investigar quantas vezes e em que momentos se deve executar a MD, a fim de introduzir memória à meta-heurística *GRASP*	1. Primeira proposta; versão 1, aplicar MD após 20% das iterações *GRASP*; versão 2, aplicar MD após 80% das iterações *GRASP*. 2. Segunda proposta, executar o algoritmo de MD três vezes, uma após 20% das iterações e as outras com 50% e 80% das iterações. 3. Terceira proposta; apresenta 3 versões, o algoritmo de mineração é aplicado quando o conjunto de elite não sofre alteração por um número de iterações igual a 5% do total (versão 1) ou 10% (versão 2) ou 20% (versão 3).	Problema de replicação de servidores para transmissão *multicast* confiável
4. Fonseca *et al*. (2009)	Introduzir memória à meta-heurística *GRASP*	1. Executar o processo de mineração de dados toda vez que o conjunto de elite (procedimento *GRASP*) se torna estável, a fim de extrair padrões frequentes mais refinados do conjunto de elite. 2. Aplicar o *GRASP* novamente a fim de construir soluções que contenham os itens mais frequentes, obtidos na fase anterior.	1. Problema de replicação de servidores para transmissão *multicast* confiável. 2. Problema das *P*-medianas.
5. Reynolds e Iglesia (2009)	Adaptar algoritmos MO para a tarefa de classificação parcial	Utilizar a meta-heurística *GRASP* apenas como parte de um algoritmo MO com o objetivo de selecionar regras para uma determinada classe de um banco de dados.	Classificação parcial

continua

TABELA 3 – TRABALHOS CORRELATOS *VERSUS* META-HEURÍSTICA PROPOSTA

conclusão

Autor(es)/ ano	Objetivo	Proposta entre *DM* e *GRASP*	Problema resolvido
6. Barbalho *et al*. (2011)	Introduzir memória à meta-heurística *GRASP*	1. Primeira proposta: GRASP-PR – aplica-se religação de caminhos após cada iteração *GRASP*, ligando-se a solução obtida da busca local com uma solução do conjunto de elite. 2. Segunda proposta: DM-GRASP-PR – executam-se *n* iterações *GRASP* para obtenção do conjunto de elite; executam-se novamente *n* iterações *GRASP*, porém com a fase de construção guiada por um padrão extraido do conjunto de elite. 3. Terceira proposta: MDM-GRASP-PR – difere da anterior pelo fato de o processo de mineração de dados ser executado toda vez que o conjunto de elite se torna estável.	2-*path network design problem*
7. Plastino *et al.* (2011)	Introduzir memória à meta-heurística *GRASP*	1. Primeira proposta: DM-GRASP – executam-se *n* iterações *GRASP* para obtenção do conjunto de elite; aplica-se MD para extrair padrões frequentes no grupo de elite; executam-se novamente *n* iterações *GRASP*, porém com a fase de construção guiada por um padrão extraido do conjunto de elite. 2. Segunda proposta: MDM-GRASP – difere da anterior pelo fato de o processo de mineração de dados ser executado toda vez que o conjunto de elite se torna estável.	Problema das *P*-medianas
8. Semaan e Ochi (2011)	Identificar a relevância das regras de associação	1. Primeira fase: formação de *itemsets.* 2. Segunda fase: construção da solução (regra de associação). 3. Avaliação e seleção das regras obtidas a partir do "*lift*".	Problema de Associação
9. Meta-heurística proposta nesta tese	Extrair regras de classificação em bases de dados	1. Primeira etapa: Pré-processamento dos dados. 2. Segunda etapa: Baseado em *GRASP* – Construção da regra de classificação e Busca Local. 3. Construção do classificador, segundo critérios de "confiança" e de "suporte".	Problema de Classificação

Na Tabela 3 se apresentam em ordem cronológica os trabalhos analisados nesta pesquisa e que utilizam a mineração de dados associada à meta-heurística *GRASP*. Na primeira coluna, listam-se os autores responsáveis pelo trabalho. Nas colunas dois e quatro se apresentam, respectivamente, o objetivo e o problema resolvido. A coluna três apresenta, de maneira sucinta, como cada trabalho relacionou técnicas de MD com *GRASP*.

Analisando a Tabela 3, nota-se que a aplicação da MD busca fornecer uma memória adaptativa à meta-heurística *GRASP* (linhas 1, 2, 3, 4, 6 e 7 da Tabela 4).

Essas hibridizações propostas consistem basicamente em: realizar determinado número de iterações *GRASP* a fim de construir um conjunto de soluções de elite; em seguida aplicar MD neste conjunto a fim de extrair padrões frequentes nas soluções que o compõem; por fim, aplicar novamente a meta-heurística *GRASP* utilizando-se dos padrões extraídos na fase anterior para direcionar a construção de novas soluções. O trabalho de Reynolds e Iglesia (2009) (linha 5) utiliza o *GRASP* como parte de um algoritmo MO com o objetivo de selecionar regras. O artigo apresentado na linha 8 aplica a meta-heurística *GRASP* a fim de extrair regras de associação, ou seja, regras que associem entre si atributos de uma mesma base de dados, e avalia a relevância dessas regras a partir da medida de interesse "*lift*".

Embora os trabalhos acima citados se utilizem da meta-heurística *GRASP*, as comparações realizadas permitem a conclusão de que a meta-heurística proposta neste trabalho diferencia-se no tipo de tarefa de *DM* executado, ou seja, classificação.

O trabalho aqui proposto vai além; viabiliza a construção de um classificador a partir das regras extraídas, ordenadas decrescentemente segundo o valor da sua confiança, com o objetivo de, além de elevada precisão preditiva, classificar todas as instâncias de uma base de dados. Assim, dado novo registro de determinada base de dados e que deva ser classificado, basta submetê-lo ao classificador que indicará, de maneira automática, a qual classe ele pertence.

3 METODOLOGIA

A Descoberta de Conhecimento em Base de Dados, foco deste trabalho, busca informações relevantes a partir de processos de mineração de dados que, por sua vez, são caracterizados pela extração automática de informações úteis a partir de bases de dados, na forma de regras e padrões. A ideia que motiva o desenvolvimento e a implementação de uma nova meta-heurística que utilize *GRASP* como ferramenta de mineração de dados justifica-se pela flexibilidade e abrangência da metodologia aqui proposta, bem como pela elevada precisão preditiva obtida nos testes aplicados.

Desta forma, o algoritmo proposto neste trabalho se baseia na meta-heurística *GRASP*, doravante denominada “meta-heurística GRASP-DM” (*GRASP* para *Data Mining,* mais especificamente, para extração de regras) A finalidade é extrair um conjunto de regras com a máxima precisão preditiva que classifique corretamente o máximo número de padrões de todas as classes que compõem uma determinada base de dados.

A inovação do algoritmo aqui proposto consiste no fato de ele fazer uso das características da meta-heurística *GRASP* para a extração de regras de classificação, tornando-a geral, para que possa ser aplicado às mais diversas bases de dados. A partir das regras geradas, constrói-se um classificador no qual o ordenamento das regras é estabelecido segundo a ordem decrescente da confiança das regras obtidas. Diante da pesquisa elaborada neste trabalho, até onde se tem conhecimento, não há indicações do uso da meta-heurística *GRASP* para problemas de extração de regras de classificação, da forma como a aqui apresentada.

A metodologia aqui proposta possui três grandes blocos segundo o processo *KDD*: pré-processamento dos dados; aplicação da meta-heurística GRASP-DM propriamente dita para extração de regras; construção do classificador. A Figura 7 apresenta as etapas da metodologia proposta, especificadas nas seções a seguir.

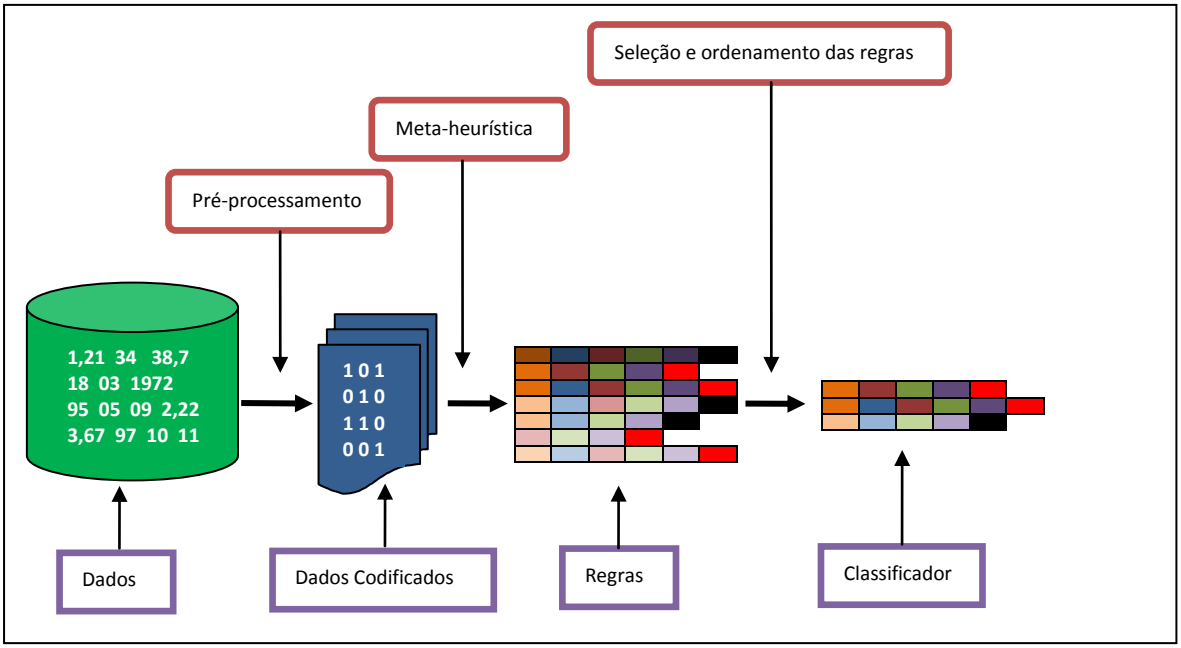

FIGURA 7 – ETAPAS DA METODOLOGIA PROPOSTA
FONTE: Autor (2014)

3.1 PRÉ-PROCESSAMENTO DOS DADOS

A meta-heurística GRASP-DM utiliza-se de valores binários no vetor de entrada. Assim, todos os atributos previsores que compõem as instâncias da base de dados, de onde serão extraídas as regras, devem ser codificados de maneira a corresponder a uma ou mais coordenadas binárias.

Os atributos que compõem as bases de dados podem ser representados por meio de variáveis de dois tipos: quantitativas ou qualitativas.

As variáveis quantitativas são aquelas que apresentam valores numéricos, podem ser medidas em uma escala e subdividem-se em discretas e contínuas. As variáveis discretas são aquelas que podem assumir apenas um número finito ou infinito contável de valores; logo, são representadas por números inteiros, como, por exemplo, número de filhos, número de televisores, etc. As variáveis contínuas assumem valores em uma escala contínua, em que valores fracionários fazem sentido, como, por exemplo, pressão arterial, peso, altura, etc.

As variáveis qualitativas são definidas por várias categorias e podem ser do tipo nominal ou ordinal. Nas variáveis nominais não existe ordenação entre as categorias, como, por exemplo, sexo de uma pessoa ou solicitou/não solicitou. Já

nas variáveis ordinais existe uma ordenação entre os valores, como, por exemplo, escolaridade (1º, 2º, 3º graus), tempo (curto, médio, longo).

Assim, cada atributo é pré-processado segundo o tipo de variável que o representa. Quando se trata de uma variável quantitativa discreta ou contínua, identifica-se o menor e o maior valor desta variável na base de dados, caracterizando o intervalo de abrangência do atributo. Este intervalo é dividido em faixas buscando manter a mesma cardinalidade dentro de cada uma delas. A Tabela 4 apresenta como exemplo a divisão em faixas do atributo “comprimento da sépala”, da base de dados íris, disponível em (*http://archive.ics.uci.edu/ml/datasets/Iris*).

TABELA 4 – EXEMPLO DE CODIFICAÇÃO DE UM ATRIBUTO QUANTITATIVO

Atributo previsor	Coordenadas	Faixas	Número de padrões em cada intervalo
A1 - Comprimento da sépala	1	4,3 <= A1<= 5,3	46
	2	5,3 < A1<= 6,2	53
	3	6,2 < A1<= 7,9	51

Como pode ser observado na Tabela 4, trata-se de um atributo quantitativo contínuo compreendido no intervalo [4,3; 7,9]. Este atributo foi dividido em três faixas e assim representado por três coordenadas. Quando o valor do “comprimento da sépala” estiver contido em uma determinada faixa, apresentará valor “1” na coordenada respectiva e “0” nas demais. Por exemplo, quando o atributo “comprimento da sépala” for igual a 5,5, a tripla binária que o representa no vetor de entrada para a meta-heurística GRASP-DM é (0 1 0).

Uma variável qualitativa nominal pode oferecer como resposta duas possibilidades: verdadeiro ou falso; pertence ou não pertence; sim ou não; etc. Desta forma, a codificação destas variáveis apresentará valor “1” se positivo (verdadeiro, pertence, sim) e “0” se negativo (falso, não pertence, não). A Tabela 5 apresenta como exemplo a codificação dos atributos: pelo, penas e ovos (base de dados zoo disponível em: *http://archive.ics.uci.edu/ml/datasets/Zoo*).

TABELA 5 – EXEMPLO DE CODIFICAÇÃO DE UM ATRIBUTO QUALITATIVO

Atributo previsor	Coordenada	Valores originais dos atributos	Número de padrões	Entrada
A1: Pelo	1	sim	43	A1 = 1
		não	58	A1 = 0
A2: Penas	2	sim	20	A2 = 1
		não	81	A2 = 0
A3: Ovos	3	sim	59	A3 = 1
		não	42	A3 = 0

A Tabela 5 apresenta três atributos qualitativos. Quando a resposta a um destes for “sim”, apresentará valor “1” na respectiva coordenada; quando for “não”, apresentará “0”. Por exemplo, quando a resposta ao atributo “penas” for igual a “sim” para um determinado padrão, a coordenada 2 deste padrão apresentará valor “1” no vetor de entrada para a meta-heurística GRASP-DM.

3.2 META-HEURÍSTICA GRASP-DM PARA EXTRAÇÃO DE REGRAS

A meta-heurística GRASP-DM é um procedimento iterativo que se divide em duas fases: construção e busca local. A Figura 8 apresenta o pseudocódigo para esta meta-heurística.

Conforme aludido no Capítulo 2, uma regra de classificação é composta de duas partes: “antecedente”, que traz as condições da regra, e “classe”, que indica a resposta a ser aprendida. Assim, o primeiro passo da meta-heurística GRASP-DM, como se pode observar na linha 1 da Figura 8, é definir o parâmetro n_A que indica o número máximo de condições que comporá o antecedente da regra de classificação.

Este número é determinado através de um teste preliminar, que se baseia na média do fator de suporte para regras em construção. Inicia-se a partir de regras com um único antecedente, ou seja, apenas uma condição no antecessor. Neste momento, geralmente, as regras tem fator de suporte elevado e confiança baixa. O número de condições do antecedente é aumentado unitariamente e á medida que se aumenta este número, a média do fator de suporte tende a diminuir e a confiança tende a aumentar. Assim, o acréscimo de condições ao antecessor é feito até que se tenha a média do fator de suporte igual ao mínimo pré-estabelecido para este

parâmetro (suporte mínimo). Neste momento, o número de condições do antecedente define o parâmetro n_A.

```
Meta-Heurística GRASP-DM
1.   nA = NúmeroCondiçõesAntecedente;
2.   α =  DeterminaTamanhoLRC;
3.   NIt = NúmeroIterações;
4.   SupMin = SuporteMínimo;
5.   ConfMin = ConfiançaMínima.
6.   MelhoresRegras = { }

7.   Para i = 1 até NIt faça

8.    Fase de Construção da Regra
9.         Regra = { };
10.        LRC(α)= { };
11.        Para j = 1 até nA faça
12.              LRC(α)= Construir a Lista Restrita de Candidatos;
13.              r = SeleçãoAleatória(LRC(α));
14.              Regra = Regra U {r};
15.        Fim Para j
16.        Retornar Regra
17.   Fim da Fase de Construção da Regra

18.   Fase de Busca Local
19.        Regra
20.        k = nA;
21.        BuscaLocal= { }
22.        Enquanto k >= 1 faça
23.           BuscaLocal = {todas as regras possíveis, combinando (k-1) a
                           (k-1) os antecessores de Regra};
24.         Para toda regra ∈ {BuscaLocal} e ∉{MelhoresRegras}faça
25.              Calcular Suporte da regra;
26.              Calcular Confiança da regra;
27.              Se Suporte da regra >= SupMin e
                     Confiança da regra >= ConfMin Então
28.              Melhores Regras = {MelhoresRegras, regra}
29.         Fim Para
30.         k = k - 1
31.        Fim Enquanto
32.   Fim Fase de Busca Local

33.  Fim Para
34.  Retornar Melhores Regras
Fim Meta-Heurística GRASP DM
```

FIGURA 8 – PSEUDOCÓDIGO DA META-HEURÍSTICA GRASP-DM
FONTE: Autor (2014)

Outros parâmetros devem ser definidos, como se pode observar nas linhas 2 a 5 da Figura 8. O parâmetro *α* vai determinar quão aleatório ou quão guloso é o algoritmo, conforme será explicitado durante a fase de construção. Quanto maior o

número de iterações (*NIt*), maior será o número de regras extraídas. O suporte mínimo (*SupMin*) exigido para se armazenar uma regra evidencia o número mínimo de padrões que se pretende cobrir com essa regra. A confiança mínima (*ConfMin*) influencia diretamente na precisão preditiva do classificador a ser construído a partir das regras armazenadas (Melhores Regras).

3.2.1 Fase de Construção da Meta-heurística GRASP-DM

O pseudocódigo da meta-heurística GRASP-DM indica que a fase de construção parte de uma regra inicial que é um conjunto vazio. A fase de construção da regra é iterativa, como se pode observar na linha 11 da Figura 8. A cada uma das n_A iterações desta fase, um elemento (condição) é acrescido à regra parcial até obter-se a regra completa, ou seja, n_A elementos no conjunto de antecedentes que classifiquem corretamente o atributo consequente (classe).

A cada iteração desta fase, os candidatos a comporem a regra são obtidos a partir do conjunto de elementos que não comprometam a viabilidade da regra – elementos que, quando inseridos à regra, deverão classificar ao menos um padrão da base de dados. Esses candidatos a comporem a regra em construção serão avaliados por uma função baseada no fator de suporte. Aqueles que satisfizerem tal condição comporão a Lista Restrita de Candidatos (LRC) (linha 12), e um elemento (atributo) desta lista será selecionado aleatoriamente (linha 13) para compor o antecedente da regra em construção (linha 14 da Figura 8). As n_A iterações são executadas até que a regra final seja obtida. O final do processo retorna, como dito anteriormente, uma regra com n_A antecedentes (linha 16).

A seguir, especifica-se como é a proposta apresentada neste trabalho para a construção da LRC (linha 12 da Figura 8).

Considere o problema de extração de regras de classificação de uma base de dados. Seja $A = \{a_1, a_2, \ldots, a_n\}$ um conjunto de elementos a serem acrescidos a uma regra. Define-se $s(a_i)$ o valor do suporte da regra após a inclusão do elemento a_i. Sejam s^{max} e s^{min}, o maior e o menor suporte das regras, respectivamente. A LRC é composta por elementos a_i pertencentes a A com os maiores suportes, de maneira que a sua inserção não destrua a viabilidade da regra. A lista fica associada ao parâmetro $\alpha \in [0, 1]$. Os elementos pertencentes à LRC devem apresentar, quando inseridos à regra, um suporte maior ou igual a um valor (Δ) pré-definido com base

no parâmetro *α*. A equação (3), a seguir, define Δ para a meta-heurística GRASP-DM.

$$\Delta = s^{\min} + \alpha(s^{\max} - s^{\min}) \quad (3)$$

Como se pode observar na equação (3), o parâmetro *α* determina o quão gulosa ou aleatória será a inserção de um novo elemento à regra durante a sua construção. Neste caso, para *α* = 0 o algoritmo é puramente aleatório, enquanto para *α* = 1 o algoritmo é puramente guloso.

3.2.2 Fase de Busca Local da Meta-heurística GRASP-DM

Nesta fase da meta-heurística GRASP-DM, o objetivo é realizar uma busca local na vizinhança da regra apresentada, a fim de buscar outras regras que apresentem grande precisão preditiva que, neste trabalho, denominaremos regras de boa qualidade. O algoritmo proposto para esta fase estabelece todas as combinações possíveis das n_A condições que compõem o antecedente da regra gerada na fase anterior. Desta forma, serão estabelecidas regras com n_A-1 elementos no antecessor, em seguida n_A-2 elementos, e assim sucessivamente até a obtenção das regras com apenas um elemento no antecessor.

O pseudocódigo (Figura 8) mostra que a fase de busca local da meta-heurística GRASP-DM parte de uma regra inicial (linha 19), obtida da fase anterior (fase de construção). O parâmetro *k* (*k* = n_A) determina a cardinalidade do antecessor da regra (linha 20). A cada iteração desta fase gera-se um conjunto de regras com n_A -1 elementos no antecessor. Cada regra gerada é avaliada segundo critérios pré-estabelecidos, ou seja, um suporte mínimo (*SupMin*) e uma confiança mínima (*ConfMin*) – conforme pode ser observado na linha 27 – de maneira que as regras que atenderem a estes critérios (Melhores Regras – linha 28) serão arquivadas.

A Tabela 6 apresenta como exemplo todas as regras obtidas na fase de busca local a partir da regra “SE (A e B e C e D) ENTÃO (S)” construída na primeira fase da meta-heurística GRASP-DM.

TABELA 6 – EXEMPLO DE BUSCA LOCAL DA META-HEURÍSTICA GRASP-DM

Nr de Antecedentes	Condições	Classe	Regras Obtidas
4	(A, B, C, D)	(S)	(A, B, C, D) ⇒ (S)
3	(A, B, C)	(S)	(A, B, C) ⇒ (S)
	(A, B, D)		(A, B, D) ⇒ (S)
	(A, C, D)		(A, C, D) ⇒ (S)
	(B, C, D)		(B, C, D) ⇒ (S)
2	(A, B)	(S)	(A, B) ⇒ (S)
	(A, C)		(A, C) ⇒ (S)
	(A, D)		(A, D) ⇒ (S)
	(B, C)		(B, C) ⇒ (S)
	(B, D)		(B, D) ⇒ (S)
	(C, D)		(C, D) ⇒ (S)
1	(A)	(S)	(A) ⇒ (S)
	(B)		(B) ⇒ (S)
	(C)		(C) ⇒ (S)
	(D)		(D) ⇒ (S)

Pode-se observar a partir da Tabela 6 que, na primeira iteração da fase de busca local do algoritmo proposto, os antecedentes (A, B, C, D) são combinados três a três. Na segunda iteração todos os antecedentes são combinados dois a dois e, por fim, cada condição apresenta-se como antecedente de uma regra. Desta forma, a partir de uma regra com quatro condições no antecedente obtida na fase de construção, serão obtidas outras 14 regras na fase de busca local.

Cabe ressaltar que todas as regras obtidas, tanto na fase de construção quanto na fase de busca local, serão avaliadas segundo seu suporte e sua confiança. As regras que apresentarem suporte e confiança superiores aos mínimos pré-estabelecidos (linha 27) serão arquivadas (Melhores Regras – linha 28).

Ao final da fase de busca local encerra-se uma iteração da meta-heurística proposta (linha 32). O procedimento inicia-se novamente, primeiramente a fase de construção, depois a fase de busca local, mais regras armazenadas no conjunto “Melhores Regras”, e assim sucessivamente até que se atinja o número de iterações (*NIt*).

Ao final do procedimento apresentado, a meta-heurística GRASP-DM armazenou um conjunto de regras (Melhores Regras – linha 34) em que cada uma delas apresenta fator de suporte e confiança maiores ou iguais aos mínimos pré-

estabelecidos. A partir deste conjunto de regras se inicia o terceiro grande bloco do procedimento desenvolvido neste trabalho, a construção do classificador.

3.3 CONSTRUÇÃO DO CLASSIFICADOR

Do conjunto de regras (Melhores Regras) obtido a partir da meta-heurística GRASP-DM, constrói-se um classificador, ou seja, um conjunto finito e sequencial de regras utilizado para classificar novos padrões.

A quantidade de regras extraídas de uma base de dados a partir da meta-heurística proposta no presente trabalho depende dos parâmetros fornecidos, ou seja, depende do número de iterações (*NIt*), do número de condições do antecedente (n_A), do suporte mínimo (*SupMin*) e da confiança mínima (*ConfMin*). Ao se variarem tais parâmetros, obtém-se um número maior ou menor de regras de classificação para a base de dados em uso. Desta forma, tem-se que a meta-heurística GRASP-DM é parametrizável, possibilitando a obtenção de um número grande de regras (Melhores Regras). Assim, nem todas as regras geradas serão utilizadas na construção do classificador.

Necessita-se, então, elaborar um processo de seleção das regras obtidas na meta-heurística proposta para a construção do classificador. Diante do objetivo de construir um classificador com elevada precisão preditiva, este trabalho propõe a seleção das regras segundo a ordem decrescente da sua confiança. A regra de maior confiança será a primeira a compor o classificador, a próxima regra será aquela que apresenta a segunda maior confiança e assim sucessivamente. Quando duas ou mais regras apresentam a mesma confiança, o critério de desempate para fins de apresentação das regras adotado neste trabalho foi o fator de suporte, ou seja, a regra que apresentar maior fator de suporte será apresentada antes das demais que apresentam a mesma confiança. Persistindo o empate, ou seja, para duas ou mais regras que apresentem a mesma confiança e o mesmo suporte, a sequência da apresentação obedecerá a cardinalidade do antecedente da regra, ou seja, a regra que apresentar menor número de atributos no antecedente será apresentada primeiramente.

3.4 AVALIAÇÃO DA META-HEURÍSTICA GRASP-DM

Para avaliação da meta-heurística GRASP-DM foi adotado o procedimento *k-fold-cross-validation* (validação cruzada com *k* = 10) cujo objetivo é analisar a capacidade de generalização de um conjunto de regras, a partir de uma base de dados, ou seja, verificar quão acurado é este conjunto de regras quando submetido a novos dados.

Segundo o processo de validação adotado, a base de dados é dividida em 10 partes (*fold*) estratificadas - cada parte deve conter uma amostra proporcional das classes que compõem a base de dados. Nove partes são utilizadas no treinamento da meta-heurística e uma é utilizada como grupo de teste.

O grupo de treinamento é submetido à meta-heurística GRASP-DM, as regras geradas são avaliadas e aquelas que atenderem os critérios de avaliação são arquivadas no conjunto “Melhores Regras”. A construção do classificador começa, conforme especificado na seção anterior, a partir da regra (pertencente ao conjunto Melhores Regras) que possua maior confiança.

Todos os padrões são apresentados individualmente a esta regra. Quando os atributos de uma determinada instância satisfizerem as condições (antecedentes) desta regra, a instância é classificada. Se a classe da instância for igual à classe da regra, esta instância foi classificada corretamente; caso contrário, a instância foi classificada erroneamente. Todas as instâncias classificadas (correta ou incorretamente) são retiradas do conjunto de dados.

Continuando a construção do classificador, a segunda regra é adicionada – segundo o critério de confiança e suporte – as instâncias classificadas são retiradas do conjunto de dados. Este processo continua até que todas as instâncias do grupo de treinamento sejam classificadas.

Ao final deste processo, tem-se o conjunto de regras apresentado ao grupo de treinamento, ou seja, tem-se o classificador para este *fold*. Em seguida, o conjunto de teste é submetido a este classificador e são verificadas quantas instâncias deste conjunto foram classificadas corretamente e quantas foram classificadas incorretamente.

Este processo é repetido 10 vezes, de maneira que em cada uma destas repetições um conjunto de dados diferente será utilizado como grupo de testes. No final das repetições, tem-se toda a base de dados submetida como grupo de teste. A

precisão preditiva será obtida dividindo-se o número de instâncias classificadas corretamente pelo número total de instâncias da base de dados.

A Figura 9 a seguir representa o procedimento *k-fold-cross-validation* (*k* = 10). Observe-se que as amostras devem ser sempre estratificadas, ou seja, o número de instâncias de cada amostra, contidas no conjunto de treinamento e de teste, deverá ser proporcional ao número total de instâncias de cada amostra.

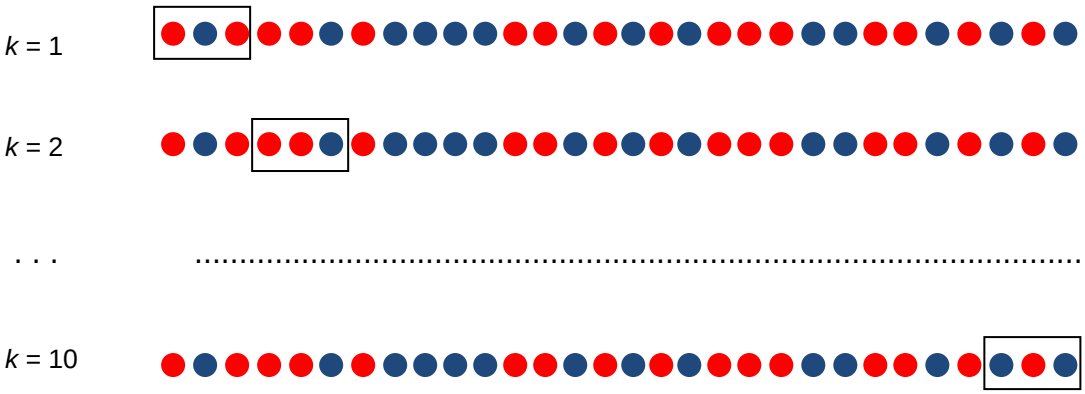

FIGURA 9 – PROCEDIMENTO *k-fold-cross-validation*
FONTE: Autor (2014)

4 RESULTADOS OBTIDOS

Calcado no objetivo de apresentar uma meta-heurística baseada em *GRASP* para extração de regras de classificação (meta-heurística GRASP-DM), bem como a utilização destas regras na construção de um classificador e, ainda, objetivando verificar sua versatilidade, esta meta-heurística foi aplicada a sete bases de dados distintas:

- Base de dados da justiça do trabalho, composta de 100 instâncias, 10 atributos e três classes;
- Base de dados wine, composta de 178 padrões, 13 atributos e três classes;
- Base de dados zoo, composta de 101 amostras, 16 atributos e sete classes;
- Base de dados íris, composta de 150 amostras, quatro atributos e três classes;
- Base de dados médico, constituída de 118 amostras, 14 atributos e duas classes;
- Base de dados *pima indians diabetes*, constituída por 768 padrões, oito atributos e duas classes;
- Base de dados *balance scale weight & distance database*, com 625 amostras, quatro atributos e três classes.

A aplicação da meta-heurística GRASP-DM, bem como os resultados das três primeiras bases (justiça do trabalho, wine e zoo) são apresentados a seguir, enquanto as demais (íris, médico, *pima indians diabetes* e *balance scale weight & distance database*) se encontram no Apêndice.

Cabe ressaltar que a metodologia empregada em todas as bases de dados, conforme detalhado no capítulo 3, possui três grandes blocos segundo o processo *KDD*: pré-processamento dos dados; aplicação da meta-heurística GRASP-DM propriamente dita para extração de regras; construção do classificador. Cada um destes blocos é detalhado a seguir.

4.1 BASE DE DADOS DA JUSTIÇA DO TRABALHO

Esta base de dados foi extraída da 1ª Vara da Justiça do Trabalho de São José dos Pinhais, Paraná, no período de setembro a novembro de 2006, e apresenta dados acerca de processos que já tiveram suas sentenças emitidas entre os anos de 1998 e 2005. É composta de 100 processos (instâncias ou padrões), cada qual com 10 atributos previsores, cujo objetivo é classificar cada instância quanto ao tempo de duração do processo em três classes distintas: tempo longo, tempo médio e tempo curto.

Os atributos classificadores foram definidos juntamente com um especialista da área (juiz do trabalho) e listam-se a seguir:

- Objeto do Processo: corresponde às solicitações feitas pelo reclamante. Estas, dentre outros, podem ser do tipo: falta de registro em carteira profissional, horas extras, fundo de garantia por tempo de serviço, verbas rescisórias, seguro-desemprego, vale-transporte, adicional de insalubridade, multa do Art. 477, adicional noturno, diferenças salariais, multa do Art. 467 e indenização por danos morais.
- Salário do Reclamante: refere-se ao último salário recebido pelo autor do processo.
- Rito: trata do tipo de rito a ser seguido no processo. O rito pode ser de dois tipos: de trabalho (RT) ou processo sumaríssimo (PS).
- Perícia: se há necessidade (ou não) de realização de alguma espécie de perícia. Neste caso, tem-se como exemplos a perícia médica ou a de periculosidade.
- Tempo de Serviço: é dado em meses pela diferença entre as datas de dispensa e de admissão.
- Acordo: quando as partes (reclamante e reclamado) entram em acordo antes do julgamento do pedido.
- Profissão: trata das funções exercidas pelo reclamante. Dividiu-se este atributo em duas partes: setor, que pode ser comércio, indústria e serviço; cargo, que pode ser direção e execução.

- Recurso Ordinário (RO): quando uma das partes (reclamante ou reclamado) não concorda com a sentença emitida pelo juiz e solicita RO ao TRT.
- Recurso de Revista (RR): quando uma das partes (reclamante ou reclamado) não concorda com o acórdão emitido pelo TRT e solicita RR ao TST.
- Número de Audiências: trata do número de audiências necessárias para que o juiz emita a sentença.

4.1.1 Pré-processamento dos dados da base justiça do trabalho

Cada um dos 10 atributos citados na seção anterior foi "codificado" de maneira a corresponder a uma ou mais coordenadas binárias (PAVANELLI, 2007; BAESENS *et al*., 2003). O atributo "Objeto do Processo", por exemplo, que corresponde a uma das solicitações realizadas pelo autor do processo (atributo nominal), apresentará o valor "1", se for solicitada; valor "0", em caso de não solicitação. A Tabela 7, a seguir, mostra a codificação desse atributo, que terá um total de 12 entradas.

TABELA 7 – CODIFICAÇÃO DO ATRIBUTO OBJETO DO PROCESSO

Objeto do Processo – A1 (atributo nominal)	Coordenada	Valores originais dos Atributos	Número de padrões em cada intervalo	Entrada
Falta de Registro em CPTS.	1	sim	11	1
		não	89	0
Horas Extras.	2	sim	87	1
		não	13	0
Fundo de Garantia por Tempo de Serviço	3	sim	52	1
		não	48	0
Verbas Rescisórias	4	sim	65	1
		não	35	0
Seguro-desemprego	5	sim	19	1
		não	81	0
Vale-transporte	6	sim	13	1
		não	87	0

continua

TABELA 7 – CODIFICAÇÃO DO ATRIBUTO OBJETO DO PROCESSO

conclusão

Objeto do Processo – A1 (atributo nominal)	Coordenada	Valores originais dos Atributos	Número de padrões em cada intervalo	Entrada
Adicional Insalubridade	7	sim	21	1
		não	79	0
Multa Art. 477	8	sim	55	1
		não	45	0
Adicional Noturno	9	sim	4	1
		não	96	0
Diferenças Salariais	10	sim	28	1
		não	72	0
Multa Art. 467	11	sim	28	1
		não	72	0
Danos Morais	12	sim	15	1
		não	85	0

O atributo "Salário do Reclamante" foi dividido em três faixas. As coordenadas binárias que o representam foram definidas conforme a Tabela 8, apresentando, assim, três coordenadas no vetor de entrada do procedimento *GRASP*.

TABELA 8 - CODIFICAÇÃO DO ATRIBUTO SALÁRIO DO RECLAMANTE

Atributo	Valores originais dos Atributos	Intervalos	Número de padrões em cada intervalo	Coordenadas		
	Faixas	Salário em reais		13	14	15
Salário do Reclamante - A2	1	Sal ≤ 450	29	1	0	0
	2	450 < Sal ≤ 800	26	0	1	0
	3	Sal > 800	45	0	0	1

Quando o atributo (A3) "Rito" for do tipo RT, apresentará coordenada (16) com valor "1"; quando for do tipo PS, apresentará valor "0". Se for necessária a execução de qualquer tipo de perícia (A4), este dado apresentará entrada (17) igual a "1" no vetor de entrada; caso contrário, apresentará valor "0".

Analogamente ao tratamento dispensado ao atributo “Salário do Reclamante”, o atributo ordinal "Tempo de Serviço", foi dividido em faixas. Pode-se observar, na Tabela 9, que este apresenta quatro coordenadas no vetor de entradas.

TABELA 9 - CODIFICAÇÃO DO ATRIBUTO TEMPO DE SERVIÇO

Atributo	Valores originais dos Atributos	Intervalos	Número de padrões em cada intervalo	Coordenadas			
	Faixa	Em meses		18	19	20	21
Tempo de Serviço (Tp Sv) – A5 (atributo ordinal)	1	Tp Sv ≤ 6	16	1	0	0	0
	2	6 < Tp Sv ≤ 14	14	0	1	0	0
	3	14 < Tp Sv ≤ 28	27	0	0	1	0
	4	Tp Sv > 28	47	0	0	0	1

Quando as partes (reclamante e reclamado) entram em acordo (A6) antes do julgamento do pedido, esta entrada apresentará o valor "1" (Coordenada 22 = 1); caso contrário, apresentará valor "0" (Coordenada 22 = 0). Conforme comentado anteriormente, o atributo "Profissão" (A7) foi dividido em setor e cargo. Por se tratar de uma variável nominal, contará com quatro coordenadas no vetor entrada de dados, como se observa na Tabela 10, a seguir.

TABELA 10 - CODIFICAÇÃO DO ATRIBUTO PROFISSÃO

Atributo	Setor do Atributo	Cargo do Atributo	Número de padrões em cada Setor/Cargo	Coordenadas			
	Setor	Cargo		23	24	25	26
Profissão-A7 (atributo nominal)	Comércio, Indústria e Serviço	Direção	5	0	0	0	1
	Comércio	Execução	13	0	0	1	0
	Indústria	Execução	39	0	1	0	0
	Serviço	Execução	43	1	0	0	0

Tanto o "Recurso Ordinário" (A8) quanto o "Recurso de Revista" (A9), se forem solicitados, apresentarão valor "1" nas respectivas coordenadas do vetor entrada de dados; caso contrário, apresentarão valor "0".

O tratamento dado ao atributo "Número de Audiências" (A10) foi o mesmo dado aos do "Tempo de Serviço" e do "Salário do Reclamante", uma vez que se trata de uma variável quantitativa, neste caso, discreta. Este dado apresenta quatro coordenadas no vetor de entrada, conforme se vê na Tabela 11, a seguir.

TABELA 11 - CODIFICAÇÃO DO ATRIBUTO NÚMERO DE AUDIÊNCIAS

Atributo	Valores originais dos Atributos	Intervalos	Número de padrões em cada intervalo	Coordenadas			
	Faixa	Audiências		29	30	31	32
Número de Audiências – A10	1	1	48	1	0	0	0
	2	2	33	0	1	0	0
	3	3	14	0	0	1	0
	4	≥ 4	5	0	0	0	1

Assim sendo, obtém-se um vetor com 32 coordenadas binárias, que correspondem aos 10 atributos definidos para cada processo. Como foram extraídos dados de 100 processos, a matriz de entrada de dados da meta-heurística proposta é da ordem de (100 x 32).

Os atributos classe, ou seja, os tempos de duração dos processos que variaram – no caso estudado, de 2 a 94 meses – foram divididos em três intervalos, conforme apresentado na Tabela 12.

TABELA 12 - CODIFICAÇÃO DO ATRIBUTO CLASSE: TEMPO DE DURAÇÃO DO PROCESSO TRABALHIISTA

Atributo Classe	Valores originais das Classes	Intervalos em meses	Número de padrões em cada Classe
Tempo de Processo	Tempo Curto	Tempo<=12	27
	Tempo Médio	12 < Tempo <= 25	36
	Tempo Longo	Tempo > 25	37

4.1.2 Aplicação da meta-heurística GRASP-DM para a base de dados da justiça do trabalho

Na sequência, procedeu-se à aplicação da meta-heurística GRASP-DM. Para realizar esta implementação foi desenvolvido um programa no *Software Visual Studio* 2012. Os parâmetros envolvidos no procedimento são: o critério de parada, o qual ficou estabelecido de acordo com o número de iterações igual a 100; a cardinalidade da LRC (α), que ficou definida como 0,5 (α = 0,5), ou seja, um “meio termo” entre a total aleatoriedade e o procedimento puramente guloso; a confiança

mínima, que ficou estipulada como 0,5 (*ConfMin* = 0,5); o suporte mínimo, pré-estabelecido como 0,05 (*SupMin* = 0,05).

Também é importante definir o número máximo de antecedentes da regra. Cabe ressaltar que, à medida que se aumenta a cardinalidade do antecedente, o fator suporte da regra tende a diminuir, como pode ser observado no Gráfico 1.

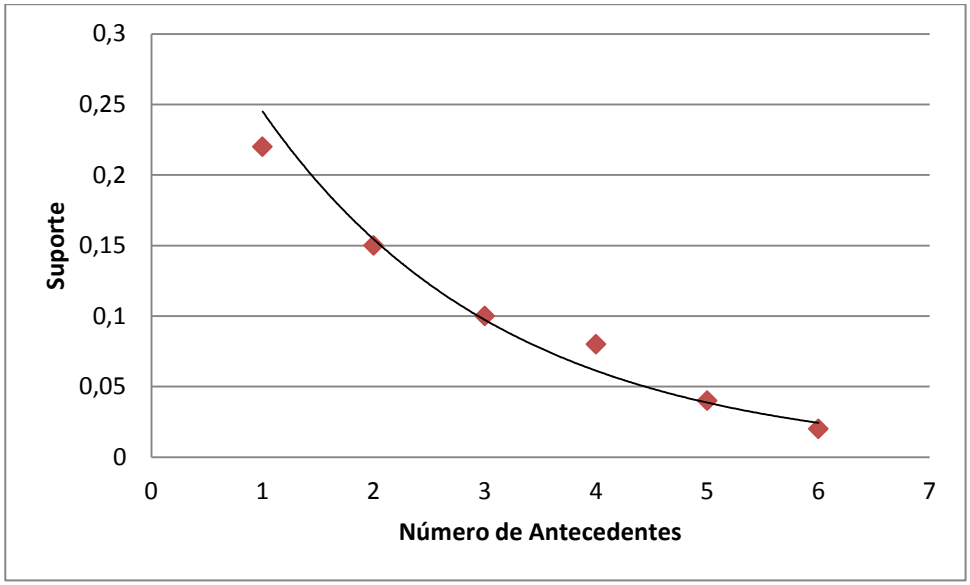

GRÁFICO 1 - RELAÇÃO ENTRE O SUPORTE E O NÚMERO DE ANTECEDENTES DA BASE DE DADOS DA JUSTIÇA DO TRABALHO

Como se pode observar no Gráfico 1, o fator de suporte, para este exemplo, é igual a 22% quando a regra é composta por apenas um antecedente. Este valor decresce com uma aproximação exponencial à medida que se aumenta o número de antecedentes da regra, de maneira que, com seis antecedentes, esta regra apresenta um fator de suporte de apenas 2%.

Observada esta condição, foram realizados testes, determinando-se, nesta base de dados, o valor máximo para o número de antecedentes igual a quatro (n_A = 4). É oportuno relembrar que a regra gerada na fase de construção apresentará o antecedente com número de atributos igual a *k*; na fase de busca local, porém, serão estabelecidas regras com n_A -1, n_A -2, ... , 1, até se chegar a “1” único atributo preditivo no antecedente da regra e desde que atendam aos critérios de suporte e de confiança mínimos.

Durante as 100 iterações do procedimento de construção e busca local, o algoritmo armazenou todas as regras obtidas que satisfaziam as condições pré-

estabelecidas de suporte e de confiança mínimos. Na construção do classificador, devem-se apresentar as regras segundo uma determinada sequência.

Buscando aumentar a precisão preditiva, conforme definido na seção 3, este trabalho propõe a sequência de apresentação das regras segundo ordem decrescente da confiança de cada regra.

A avaliação da meta-heurística GRASP-DM foi realizada por meio do procedimento de validação cruzada *k-fold,* com *k* = 10. Desta forma, foram construídos 10 classificadores distintos. A Tabela 13 apresenta um dos classificadores estabelecidos durante o processo de validação e aplicado ao seu conjunto de treinamento.

TABELA 13 – CLASSIFICADOR TEMPO DE PROCESSO APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (Número de audiências = 1) E (Acordo = sim) ENTÃO (Tempo Curto)	15	0	75
2	SE (Tempo de serviço >28 meses) E (Acordo = sim) E (FGTS = sim) ENTÃO (Tempo Curto).	4	0	71
3	SE (Profissão = indústria) E (Rito = RT) E (Recurso ordinário = sim) E (Horas extras = sim) ENTÃO (Tempo Longo).	9	0	62
4	SE (Profissão = indústria) E (Rito = RT) E (Horas extras = sim) E (Tempo de serviço <=6 meses) ENTÃO (Tempo Longo).	2	0	60
5	SE (Horas extras = sim) E (Rito = RT) E (Recurso ordinário = sim) E (Tempo de serviço <=6 meses) ENTÃO (Tempo Longo).	2	0	58
6	SE (Verbas rescisórias = sim) E (Rito = RT) E (Horas extras = sim) E (Recurso ordinário = sim) ENTÃO (Tempo Longo).	1	0	57
7	SE (Profissão = diretoria) E (Verbas rescisórias = sim) E (Número de audiências = 1) E (Salário >R$ 800,00) ENTÃO (Tempo Curto).	1	0	56
8	SE (Profissão = indústria) E (Recurso ordinário = sim) ENTÃO (Tempo Longo).	1	0	55

continua

TABELA 13 – CLASSIFICADOR TEMPO DE PROCESSO APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRA	Class correta	Class Errada	Restam
9	SE (FGTS = sim) E (Perícia = sim) ENTÃO (Tempo Longo).	3	0	52
10	SE (FGTS = sim) E (Multa do Art. 477 = sim) E (14 meses < Tp de serviço ≤ 28 meses) E (Número de audiências = 2) ENTÃO (Tempo Médio).	4	1	47
11	SE (Multa do Art. 477 = sim) E (Horas extras = sim) E (14 meses < Tp de serviço ≤ 28 meses) E (Salário <= R$ 450,00) ENTÃO (Tempo Médio).	2	1	44
12	SE (FGTS = sim) E (Salário > R$ 800,00) E (14 meses < Tempo de serviço ≤ 28 meses) E (Rito = RT) ENTÃO (Tempo Médio).	3	0	41
13	SE (FGTS = sim) E (Salário >R$ 800,00) E (Número de audiências = 2) E (Verbas rescisórias = sim) ENTÃO (Tempo Médio).	2	0	39
14	SE (FGTS = sim) E (Salário >R$ 800,00) E (Número de audiências = 2) E (Tempo de serviço >28 meses) ENTÃO (Tempo Médio).	1	0	38
15	SE (Horas extras = sim) E (Rito = RT) E (Número de audiências = 1) E (R$ 450,00 < Sal ≤ R$ 800,00) ENTÃO (Tempo Médio).	5	0	33
16	SE (FGTS = sim) E (Multa do Art. 477 = sim) E (Verbas rescisórias = sim) E (Número de audiências = 2) ENTÃO (Tempo Médio).	2	0	31
17	SE (Profissão = indústria) E (Verbas rescisórias = sim) ENTÃO (Tempo Longo).	3	2	26
18	SE (Profissão = diretoria) E (Horas extras = sim), ENTÃO (Tempo Médio).	2	1	23
19	SE (Recurso ordinário = sim) ENTÃO (Tempo Longo).	3	1	19
20	SE (Seguro desemprego = sim) ENTÃO (Tempo Curto).	2	2	15
21	SE (Número de audiências = 1) ENTÃO (Tempo Médio).	2	1	12

continua

TABELA 13 – CLASSIFICADOR TEMPO DE PROCESSO APLICADO AO GRUPO DE TREINAMENTO

conclusão

	REGRA	Class correta	Class Errada	Restam
22	SE (Tempo de serviço >28 meses) ENTÃO (Tempo Médio).	2	1	9
23	SE (Rito = RT) ENTÃO (Tempo Longo).	6	3	0

A partir do classificador apresentado na Tabela 13, pode-se construir a matriz de confusão.

TABELA 14 – MATRIZ DE CONFUSÃO DA BASE JUSTIÇA DO TRABALHO APLICADO AO GRUPO DE TREINAMENTO

Classe	Tempo Curto	Tempo Médio	Tempo Longo	Precisão	
				Classe	Classificador
Tempo Curto	22	2	0	22/24	
Tempo Médio	1	25	6	25/32	77/90
Tempo Longo	1	3	30	30/34	

A partir da matriz de confusão, observa-se que o classificador apresenta uma precisão preditiva de 85,6% para o grupo de treinamento.

O conjunto de teste (padrões que não foram utilizados no treinamento) deste "*fold*" foi submetido ao classificador apresentado na Tabela 14. Os resultados relevantes encontram-se na matriz de confusão, conforme a Tabela 15.

TABELA 15 – MATRIZ DE CONFUSÃO DA BASE JUSTIÇA DO TRABALHO APLICADO AO GRUPO DE TESTE

Classe	Tempo Curto	Tempo Médio	Tempo Longo	Precisão	
				Classe	Classificador
Tempo Curto	3	0	0	3/3	
Tempo Médio	0	4	0	4/4	8/10
Tempo Longo	1	1	1	1/3	

A partir da matriz de confusão exibida na Tabela 15, observa-se que o classificador apresenta uma precisão preditiva de 80% para este grupo de teste.

Ao final do processo de validação cruzada, calcula-se a taxa de erro global, que é a média das taxas de erro calculadas em cada etapa. A Tabela 16 a seguir apresenta, além desta taxa, a acurácia global (1 – erro global) bem como seu desvio padrão e sua mediana, todos baseados nos 10 testes do procedimento de validação cruzada para os grupos de treinamento e teste.

TABELA 16 – PRECISÃO PREDITIVA DA BASE JUSTIÇA DO TRABALHO

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	84%	0,009	0,83
Teste	78%	0,063	0,80

4.1.3 Comparação dos resultados obtidos pela meta-heurística GRASP-DM com a técnica de árvores de decisão para a base de dados tempo de processo

A aplicação aqui abordada visa comparar o desempenho da técnica de árvores de decisão com a meta-heurística GRASP-DM, foco deste trabalho. Buscando ampliar o horizonte de comparações, foram estabelecidos três algoritmos envolvendo a técnica de árvores de decisão a partir do *software WEKA* (*Waikato Environment for Knowledge Analysis*): BFTree, REPTree e J4.8. Cabe ressaltar que em todos os testes – tanto com a meta-heurística GRASP-DM quanto com a dos algoritmos a partir do *software WEKA* – foram utilizados os mesmos conjuntos de dados.

A Tabela 17 a seguir apresenta o número de instâncias classificadas correta e incorretamente para todos os algoritmos.

TABELA 17 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS JUSTIÇA DO TRABALHO

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	70%	66%	68%	78%
Instâncias classificadas incorretamente	30%	34%	32%	22%

Como se pode observar na Tabela 17, a meta-heurística GRASP-DM apresentou melhores resultados quando comparados aos dos algoritmos de árvore de decisão.

Visando a uma comparação entre as precisões das classificações em cada uma das classes e também do classificador de cada uma das aplicações estabelecidas, foi construída a Tabela 18, na qual se tem um melhor detalhamento dos resultados.

TABELA 18 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE JUSTIÇA DO TRABALHO

Algoritmos	Precisão			
	Tempo Curto	Tempo Médio	Tempo Longo	Classificador
BFTree	78%	58%	76%	70%
RepTree	78%	50%	73%	66%
J4.8	77%	61%	68%	68%
GRASP-DM	100%	65%	75%	78%

A partir desta tabela, foi elaborado o Gráfico 2 de comparação entre as precisões apresentadas dentro de cada classe, enfatizando a superioridade da meta-heurística GRASP-DM.

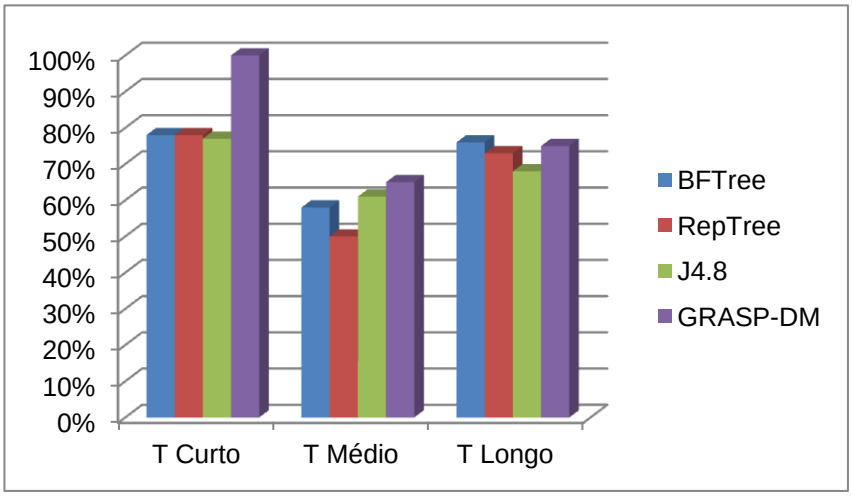

GRÁFICO 2 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS JUSTIÇA DO TRABALHO

4.2 BASE DE DADOS WINE

Trata-se de uma base de dados constituída de 178 instâncias, cada uma delas composta por 13 atributos, que correspondem aos resultados de análises químicas realizadas em três tipos de vinhos produzidos na mesma região da Itália. Esta base de dados foi extraída do *Machine Learning Repositttory* (*http://archive.ics.uci.edu/ml/datasets/Wine*). Possui 59 instâncias da primeira classe, 71 instâncias da segunda classe e 48 instâncias da terceira.

A análise envolve as quantidades de 13 constituintes encontrados em cada um dos três tipos (classes) de vinho. Assim, cada registro apresenta 13 atributos: teor de álcool, acidez, resíduo, alcalinidade, teor de magnésio, total de fenóis, flavonóides, fenóis não flavonóides, proantocianinos, intensidade da cor, coloração, OD280/OD315 de vinhos diluídos e prolina.

4.2.1 Pré-processamento dos dados da base de dados wine

Nesta base de dados, todos os atributos são representados por variáveis do tipo ordinal. Neste trabalho cada uma delas foi dividida em três intervalos, buscando manter a mesma cardinalidade em cada um deles, como se pode observar na Tabela 19.

TABELA 19 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE WINE

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A1: Teor Alcoólico	1	A1<= 12,52	60
	2	12,52 < A1 <= 13,48	59
	3	A1 > 14,83	59
A2: Acidez	4	A2<= 1,67	60
	5	1,67 < A2<= 22,55	59
	6	A2 > 22,55	59
A3: Residuo	7	A3<= 2,46	59
	8	2,46 < A3<= 3,23	58
	9	A3 > 3,23	61

continua

TABELA 19 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE WINE

conclusão

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A4: Alcalinidade	10	A4<= 18	63
	11	18 < A4<= 20,8	55
	12	A4 > 20,8	60
A5: Teor de magnésio	13	A5<= 91	58
	14	91 < A5<= 103	63
	15	A5 > 103	57
A6: Total de fenóis	16	A6<= 1,93	59
	17	1,93 < A6<= 2,61	60
	18	A6 > 2,61	59
A7: Flavonóides	19	A7<= 1,46	59
	20	1,46 < A7<= 2,64	58
	21	A7 > 2,64	61
A8:Fenóis não flavonóides	22	A8<= 0,28	57
	23	0,28 < A8<= 0,41	61
	24	A8 > 0,41	60
A9: Proantocianinos	25	A9<= 1,35	63
	26	1,35 < A9<= 1,82	56
	27	A9 > 1,82	59
A10:Intensidade da cor	28	A10<= 3,7	59
	29	3,7 < A10<= 5,64	60
	30	A10 > 5,64	59
A11: Coloração	31	A11<= 0,87	60
	32	0,87< A11<= 1,06	58
	33	A11 > 1,06	60
A12: OD280/OD315 de vinhos diluídos	34	A12<= 2,27	59
	35	2,27< A12<= 3,03	60
	36	A12 > 3,03	59
A13: Prolina	37	A13<= 550	59
	38	550 < A13<= 835	60
	39	A13 > 835	59

Assim como nas demais bases apresentadas neste trabalho, cada atributo foi "codificado" em coordenadas binárias. Conforme apresentado na Tabela 19, cada atributo foi dividido em três faixas, cada qual representada por três coordenadas binárias. Quando o atributo corresponde ao intervalo da faixa, a coordenada correspondente re**c**ebe valor “1”; caso contrário, recebe valor “0”. Se, por exemplo, o

teor alcoólico de uma determinada instância for igual a 13,0 (A1 = 13,0), as coordenadas 1 e 3 apresentarão valor “0”, enquanto a 2 apresentará valor “1”.

4.2.2 Aplicação da meta-heurística GRASP-DM para a base de dados wine

Com o objetivo de se extraírem regras que classifiquem corretamente as instâncias da base de dados, foi aplicada a meta-heurística GRASP-DM. Testes preliminares serviram de base para estabelecer o número máximo de antecedentes da regra como três.

Os parâmetros utilizados para aplicação da meta-heurística nesta base de dados foram os mesmos aplicados à base de dados tempo de processo, ou seja, o critério de parada foi de 100 iterações; alfa igual a 0,5 (α = 0,5); a confiança mínima igual a 0,5 (*ConfMin* = 0,5); o suporte mínimo igual a 0,05 (*SupMin* = 0,05).

A Tabela 20 a seguir apresenta um dos classificadores obtidos no processo de validação cruzada (*k fold*) e aplicado ao seu respectivo grupo de treinamento.

TABELA 20 – CLASSIFICADOR WINE APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (Teor Alcoólico < 12,52) E (Intensidade da cor <= 3,7) ENTÃO (Classe 2).	44	0	116
2	SE (Acidez > 22,55) E (Intensidade da cor > 5,64) ENTÃO (Classe 3).	21	0	95
3	SE (Alcalinidade< 18) E (Prolina > 835) ENTÃO (Classe 1).	32	0	63
4	SE (1,67 < Acidez <= 22,55) E (Prolina > 835) ENTÃO (Classe 1).	6	0	57
5	SE (2,27 < OD280/OD315 <= 3,03) E (Intensidade da cor > 5,64) ENTÃO (Classe 1).	3	0	54
6	SE (Coloração<= 0,87) E (OD280/OD315 <= 2,27) E (Intensidade da cor > 5,64) ENTÃO (Classe 1).	8	0	46

continua

TABELA 20 – CLASSIFICADOR WINE APLICADO AO GRUPO DE TREINAMENTO

conclusão

	REGRA	Class correta	Class Errada	Restam
7	SE (Flavonóides <= 1,46) E (Coloração<= 0,87) E (550 < Prolina <= 835) ENTÃO (Classe 3).	7	0	39
8	SE (Teor de magnésio <= 91) E (1,46 < Flavonóides <= 2,64) ENTÃO (Classe 2).	4	0	35
9	SE (Prolina <=550) E (Intensidade da cor <= 3,7) ENTÃO (Classe 2).	6	0	29
10	SE (Intensidade da cor <= 3,7) E (Teor de magnésio <= 91) ENTÃO (Classe 2).	1	0	28
11	SE (3,7< Intensidade da cor <= 5,64) E (Prolina > 835) ENTÃO (Classe 1).	6	0	22
12	SE (Intensidade da cor <= 3,7) ENTÃO (Classe 2).	2	0	20
13	SE (OD280/OD315 <= 2,27) E (Coloração<= 0,87) ENTÃO (Classe 3).	3	0	17
14	SE (Alcalinidade <= 18) E (Teor Alcoólico > 14,83) E (Flavonóides > 2,64) ENTÃO (Classe 1).	3	0	14
15	SE (Acidez > 22,55) E (Alcalinidade> 20,8) E (Flavonóides <= 1,46) ENTÃO (Classe 3).	2	0	12
16	SE (Teor Alcoólico < 12,52) ENTÃO (Classe 2).	5	1	6
17	SE (OD280/OD315 > 3,3) E (3,7< Intensidade da cor<= 5,64) ENTÃO (Classe 1).	2	0	4
18	SE (Prolina <=550) ENTÃO (Classe 2).	2	1	1
19	SE (3,7< Intensidade da cor <= 5,64) ENTÃO (Classe 1).	1	0	0

A partir do classificador apresentado na Tabela 20, pode-se montar a Matriz de confusão.

TABELA 21 – MATRIZ DE CONFUSÃO DA BASE WINE APLICADO AO GRUPO DE TREINAMENTO

Classe	Classe 1	Classe 2	Classe 3	Precisão	
				Classe	Classificador
Classe 1	54	0	0	54/54	
Classe 2	0	64	0	64/64	158/160
Classe 3	0	2	40	40/42	

A partir da matriz de confusão constante na Tabela 21, observa-se que o classificador apresenta uma precisão preditiva de 98,8% para este grupo de treinamento.

Quando submetido ao classificador explicitado na Tabela 20, o conjunto de teste deste "*fold*" exibe os resultados conforme matriz de confusão apresentada a seguir.

TABELA 22 – MATRIZ DE CONFUSÃO DA BASE WINE APLICADO AO GRUPO DE TESTE

Classe	Classe 1	Classe 2	Classe 3	Precisão	
				Classe	Classificador
Classe 1	5	0	0	5/5	
Classe 2	1	6	0	6/7	16/18
Classe 3	1	0	5	5/6	

A partir da matriz de confusão da Tabela 22 acima observa-se que este classificador apresenta uma precisão preditiva de 88,9% para o grupo de teste.

Os dados relevantes acerca do processo de validação cruzada são apresentados na Tabela 23, a seguir.

TABELA 23 – PRECISÃO PREDITIVA DA BASE WINE

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	98%	0,009	0,98
Teste	94%	0,037	0,944

4.2.3 Comparação dos resultados obtidos pela meta-heurística GRASP-DM com a técnica de árvores de decisão para a base de dados wine

Analogamente à base de dados tempo de processo, foram comparados os desempenhos da técnica de árvores de decisão – algoritmos BFTree, REPTree e J4.8 – com a meta-heurística GRASP-DM. A Tabela 24, a seguir, apresenta o número de instâncias classificadas correta e incorretamente para todos os algoritmos.

TABELA 24 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS WINE

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	89,9%	94,4%	93,8%	94,4%
Instâncias classificadas incorretamente	10,1%	5,6%	6,2%	5,6%

Como se pode observar a partir da Tabela 24, a meta-heurística GRASP-DM apresentou melhores resultados quando comparados aos BFTree e J4.8.

A Tabela 25, a seguir, apresenta as comparações entre as precisões em cada uma das classes e também do classificador de cada uma das aplicações estabelecidas.

TABELA 25 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE WINE

Algoritmos	Precisão			
	Classe 1	Classe 2	Classe 3	Classificador
BFTree	91,5%	90,1%	87,5%	89,9%
RepTree	94,9%	94,4%	93,4%	94,4%
J4.8	98,3%	94,4%	87,5%	93,8%
GRASP-DM	94,9%	95,8%	91,7%	94,4%

A partir da Tabela 25, foi elaborado o Gráfico 3 de comparação entre as precisões apresentadas dentro de cada classe.

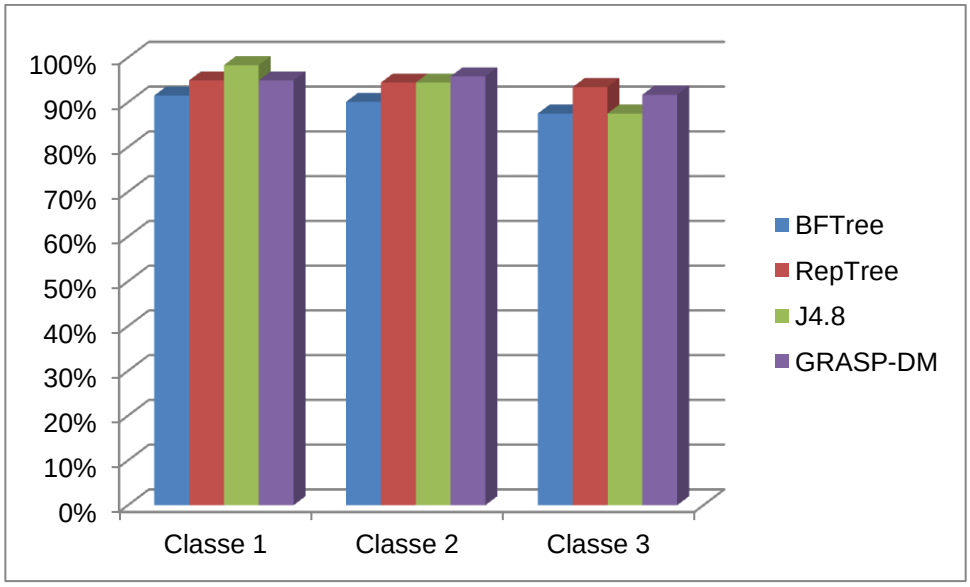

GRÁFICO 3 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS WINE

4.3 BASE DE DADOS ZOO

Buscando verificar a versatilidade da meta-heurística GRASP-DM, escolheu-se a base de dados zoo, extraída do *Machine Learning Repositttory* (*http://archive.ics.uci.edu/ml/datasets/Zoo*). Constituí-se de 101 amostras de animais, cada uma delas composta por 16 atributos previsores. Esta base se diferencia das demais por apresentar um número maior de classes: sete, as quais são apresentadas na Tabela 26.

TABELA 26 - CLASSES DA BASE ZOO

Classe	Descrição	Número de amostras
1	Mamíferos	41
2	Aves	20
3	Répteis	5
4	Peixes	13
5	Anfíbios	4
6	Insetos	8
7	Invertebrados	10

Cada amostra que classifica os animais segundo a Tabela 26 apresenta as seguintes características: pelo, penas, ovos, leite, voador, aquático, predador, dentado, coluna vertebral, respira, venenoso, barbatana, número de pernas, rabo, doméstico e dimensões aproximadas a de um gato.

4.3.1 Pré-processamento dos dados da base de dados zoo

Esta base de dados apresenta 16 atributos previsores, dos quais 15 são variáveis qualitativas nominais e uma ordinal. Os atributos nominais, que correspondem a uma das características do animal, apresentarão o valor "1" caso o animal a possua e valor "0" em caso negativo. O atributo ordinal (número de pernas) que pode apresentar os valores (0, 2, 4, 5, 6 e 8) será constituído por seis entradas binárias, ou seja, se o animal possui duas pernas, por exemplo, este atributo apresenta as seguintes coordenadas (0 1 0 0 0 0). A Tabela 27, a seguir, mostra a codificação dos atributos previsores.

TABELA 27 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE ZOO

Atributo previsor	Coordenada	Valores originais dos atributos	Número de padrões	Entrada
A1: Pelo	1	sim	43	A1 = 1
		não	58	A1 = 0
A2: Penas	2	sim	20	A2 = 1
		não	81	A2 = 0
A3: Ovos	3	sim	59	A3 = 1
		não	42	A3 = 0
A4: Leite	4	sim	41	A4 = 1
		não	60	A4 = 0
A5: Voador	5	sim	24	A5 = 1
		não	77	A5 = 0
A6: Aquático	6	sim	36	A6 = 1
		não	65	A6 = 0
A7: Predador	7	sim	56	A7 = 1
		não	45	A7 = 0
A8: Dentes	8	sim	61	A8 = 1
		não	40	A8 = 0

continua

TABELA 27 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE ZOO

conclusão

Atributo previsor	Coordenada	Valores originais dos atributos	Número de padrões	Entrada
A9: Coluna vertebral	9	sim	83	A9 = 1
		não	18	A9 = 0
A10: Respira.	10	sim	80	A10 = 1
		não	21	A10 = 0
A11: Venenoso	11	sim	8	A11 = 1
		não	93	A11 = 0
A12: Barbatana	12	sim	17	A12 = 1
		não	84	A12 = 0
A13: Número de pernas,	13	0	24	A13 = 1 0 0 0 0 0
	14	2	27	A13 = 0 1 0 0 0 0
	15	4	37	A13 = 0 0 1 0 0 0
	16	5	1	A13 = 0 0 0 1 0 0
	17	6	10	A13 = 0 0 0 0 1 0
	18	8	2	A13 = 0 0 0 0 0 1
A14: Rabo	19	sim	75	A14 = 1
		não	26	A14 = 0
A15: Doméstico	20	sim	13	A15 = 1
		não	88	A15 = 0
A16: Dimensões próximas de um gato	21	sim	44	A16 = 1
		não	57	A16 = 0

Desta forma, o vetor de entrada para a meta-heurística GRASP-DM apresentará, para esta base de dados, 21 coordenadas binárias. Como a base é composta de 101 instâncias, a matriz de dados apresenta 101 linhas e 21 colunas.

4.3.2 Aplicação da meta-heurística GRASP-DM para a base de dados zoo

Assim como nas demais bases de dados, foram realizados testes preliminares a fim de se estabelecer o número máximo de antecedentes das regras para a aplicação da meta-heurística GRASP-DM. A partir destes testes, ficou preestabelecido o número de antecedentes igual a quatro.

Devido às características desta base de dados, alguns parâmetros utilizados na aplicação da meta-heurística GRASP-DM foram alterados em relação às bases

de dados analisadas anteriormente. Manteve-se o critério de parada com 100 iterações e alfa igual a 0,5 (α = 0,5). O suporte mínimo foi alterado para 0,02 (*SupMin* = 0,02), uma vez que algumas classes possuem um número reduzido de instâncias. A “classe 5”, por exemplo, possui apenas quatro instâncias; desta forma, uma regra que classifique corretamente toda esta classe apresentará um fator de suporte igual a 0,04, ou seja inferior ao do suporte mínimo (*SupMin* = 0,05) adotado nas bases anteriores. A confiança mínima também foi reduzida para 0,2 (*ConfMin* = 0,2) a fim de apresentar regras que classifiquem todas as instâncias. A partir destas simples alterações, a meta-heurística proposta pode adequar-se facilmente a esta base de dados, explicitando sua versatilidade.

A partir das regras obtidas durante as fases de construção e de busca local, nas 100 iterações efetuadas, foi estabelecido para cada etapa do processo validação cruzada (*k fold; k* = 10) um classificador cujas regras se apresentam segundo a ordem decrescente da sua confiança. A Tabela 28, a seguir, apresenta um dos classificadores obtidos no processo aplicado ao seu respectivo grupo de treinamento.

TABELA 28 – CLASSIFICADOR ZOO APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (Leite = sim) ENTÃO (Mamífero).	37	0	54
2	SE (Penas = sim) ENTÃO (Ave).	18	0	36
3	SE (Número de pernas = 4) E (Coluna vertebral = sim) E (Aquático = sim) ENTÃO (Anfíbio).	3	0	33
4	SE (Número de pernas =6) E (Respiração = sim) ENTÃO (Inseto).	7	0	26
5	SE (Ovos = sim) E (Aquático = sim) E (Predador = sim) E (Número de pernas = 6) ENTÃO (Invertebrados).	2	0	24
6	SE (Barbatana = sim) ENTÃO (Peixe).	12	0	12
7	SE (Rabo = sim) ENTÃO (Répteis).	3	0	9
8	SE (Ovos = sim) ENTÃO (Invertebrados).	6	1	2
9	SE (Predador = sim) E (Rabo = sim) ENTÃO (Invertebrados).	1	1	0

A partir do classificador apresentado na Tabela 28, pode-se montar a matriz de confusão.

TABELA 29 – MATRIZ DE CONFUSÃO DA BASE ZOO APLICADO AO GRUPO DE TREINAMENTO

Classe	Mamíferos	Aves	Répteis	Peixes	Anfíbios	Insetos	Invertebrados	Precisão	
								Classe	Classificador
Mamíferos	37	0	0	0	0	0	0	37/37	
Aves	0	18	0	0	0	0	0	18/18	
Répteis	0	0	3	0	0	0	2	3/5	
Peixes	0	0	0	12	0	0	0	12/12	89/91
Anfíbios	0	0	0	0	3	0	0	3/3	
Insetos	0	0	0	0	0	7	0	7/7	
Invertebrados	0	0	0	0	0	0	9	9/9	

A matriz de confusão exibida na tabela acima indica que o classificador apresenta uma precisão preditiva de 97,8% para o grupo de treinamento.

O conjunto de teste deste "*fold*" foi submetido ao classificador apresentado na Tabela 28 acima e a precisão deste conjunto pode ser observada na matriz de confusão, conforme Tabela 30 a seguir.

TABELA 30 – MATRIZ DE CONFUSÃO DA BASE ZOO APLICADO AO GRUPO DE TESTE

Classe	Mamíferos	Aves	Répteis	Peixes	Anfíbios	Insetos	Invertebrados	Precisão	
								Classe	Classificador
Mamíferos	4	0	0	0	0	0	0	4/4	
Aves	0	2	0	0	0	0	0	2/2	
Répteis	-	-	-	-	-	-	-	-	
Peixes	0	0	0	1	0	0	0	1/1	10/10
Anfíbios	0	0	0	0	1	0	0	1/1	
Insetos	0	0	0	0	0	1	0	1/1	
Invertebrados	0	0	0	0	0	0	1	1/1	

Como pode ser observado na matriz de confusão apresentada na Tabela 30, este classificador exibe uma precisão preditiva de 100%, para seu grupo de teste.

Cabe ressaltar que, durante o processo de validação cruzada, cada instância é apresentada – em um dos *k-fold* – como integrante do grupo de teste. Assim, ao se agruparem os 10 grupos de testes do processo de validação, tem-se a base de dados completa apresentada como teste. A Tabela 31, a seguir, apresenta os dados estatísticos acerca da acurácia da classificação dos 101 padrões desta base quando estes compunham os grupos de teste.

TABELA 31 – PRECISÃO PREDITIVA DA BASE ZOO

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	98,2%	0,006	0,978
Teste	98%	0,042	1

4.3.3 Comparação dos resultados obtidos pela meta-heurística GRASP-DM com a técnica de árvores de decisão para a base de dados zoo

Objetivando comparar a precisão da meta-heurística GRASP-DM, a base de dados zoo foi submetida a três algoritmos de árvores de decisão (BFTree, REPTree e J4.8). Ressalta-se que foi utilizado o mesmo processo de validação (*k-fold*, com *k* = 10) tanto na meta-heurística GRASP-DM quanto nos algoritmos de árvore de decisão. A Tabela 32, a seguir, apresenta o número de instâncias da base zoo classificadas correta e incorretamente para todos os algoritmos.

TABELA 32 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS ZOO

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	40,6%	40,6%	92,1%	98%
Instâncias classificadas incorretamente	59,4%	59,4%	7,9%	2%

Como se pode observar na Tabela 32, a meta-heurística GRASP-DM apresentou, para esta base de dados, resultados muito superiores aos obtidos pelos algoritmos de árvore de decisão.

A seguir, são apresentadas as comparações entre as precisões em cada uma das classes e também do classificador de cada uma das aplicações estabelecidas.

TABELA 33 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE ZOO

Algoritmos	Precisão							
	Mamí-feros	Aves	Répteis	Peixes	Anfíbios	Insetos	Inverte-brados	Classifi-cador
BFTree	100%	0	0	0	0	0	0	41%
RepTree	100%	0	0	0	0	0	0	41%
J4.8	100%	100%	60%	100%	75%	62,5%	80%	92,1%
GRASP-DM	100%	100%	60%	100%	100%	100%	100%	98%

Como se pode observar na Tabela 33, a meta-heurística GRASP-DM apresentou resultados superiores aos de todos os demais algoritmos, não somente no classificador, mas também em três (anfíbios, insetos e invertebrados) das sete classes. Nas demais classes (mamíferos, aves, répteis e peixes) a meta-heurística aqui proposta apresenta precisões preditivas iguais ou superiores às dos demais algoritmos comparativos. A seguir apresenta-se o Gráfico 4, elaborado a partir das precisões das sete classes que compõem esta base de dados.

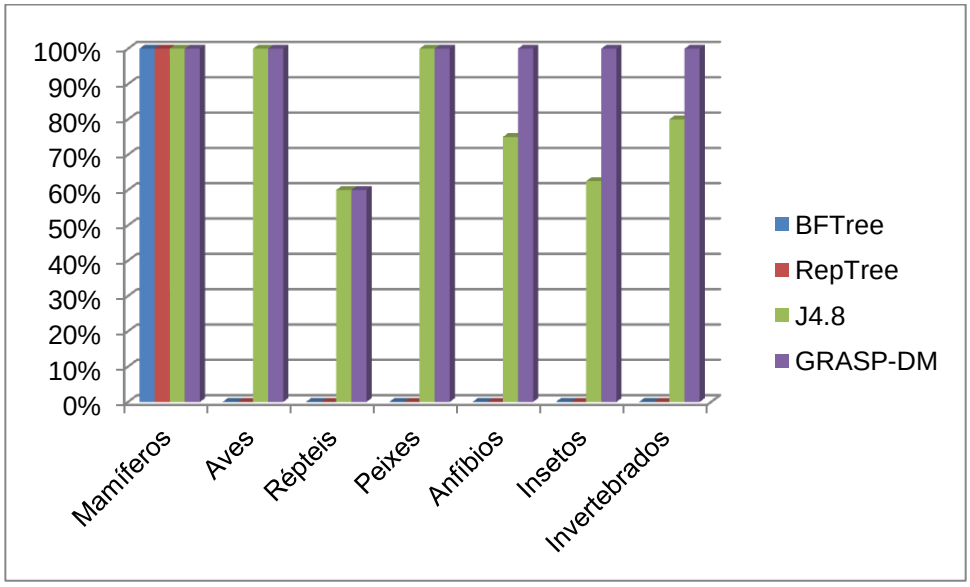

GRÁFICO 4 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS ZOO

4.4 ANÁLISE DA APLICAÇÃO DA META-HEURÍSTICA GRASP-DM

Conforme aludido no início desta seção, a meta-heurística GRASP-DM foi aplicada a sete bases de dados. Nos itens 4.1, 4.2 e 4.3 foi apresentado detalhadamente o procedimento adotado neste trabalho para as bases de dados justiça do trabalho, wine e zoo. Nesta subseção, a partir da Tabela 34, serão apresentados de maneira sucinta os dados relevantes referentes às sete bases analisadas.

Pela Tabela 34, representada pelo Gráfico 5, verifica-se que a meta-heurística GRASP-DM apresentou, para seis bases de dados precisão preditiva superior à dos algoritmos de árvore de decisão. Em apenas uma das bases (wine) a acurácia obtida pela proposta deste trabalho foi igual à dos resultados obtidos pelos algoritmos comparativos. Estes resultados demonstram a superioridade (para as bases de dados analisadas) quanto à precisão preditiva da meta-heurística GRASP-DM em relação à dos demais algoritmos utilizados.

No Apêndice encontra-se o detalhadamente do procedimento proposto neste trabalho para as bases de dados íris, médico, *pima indians diabetes e balance scale weight & distance database*, conforme apresentado para as demais bases de dados nos itens 4.1, 4.2 e 4.3.

TABELA 34 – RESULTADOS DOS ALGORITMOS APLICADOS ÀS SETE BASES DE DADOS

Bases	Nr de Padrões	Nr de Atributos	Nr de Classes	Nr de Regras				Acurácia			
				BF Tree	Rep Tree	J4.8	GRASP-DM	BF Tree	Rep Tree	J4.8	GRASP-DM
Justiça	100	10	3	14	3	15	23	70%	66%	68%	78%
Wine	178	13	3	4	9	5	19	90%	94%	93%	94%
Zoo	101	16	7	1	1	9	9	41%	41%	92%	98%
Íris	150	4	3	6	5	5	4	95%	94%	96%	97%
Médico	118	14	2	7	7	7	29	75%	77%	73%	85%
Pima	768	8	2	3	49	20	81	74%	75%	74%	76%
Balance	625	4	3	81	59	52	213	79%	77%	77%	90%

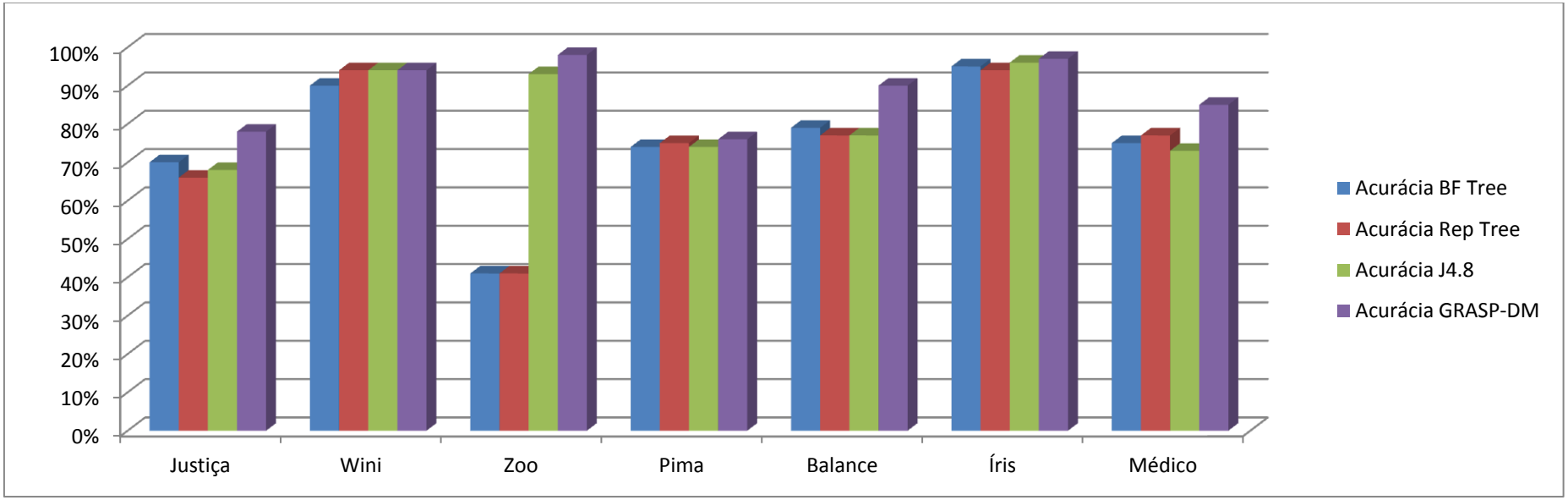

GRÁFICO 5 – COMPARATIVO ENTRE AS PRECISÕES DOS CLASSIFICADORES PARA SETE BASES DE DADOS.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Como comentado na introdução desta Tese, este trabalho aborda o processo de Descoberta de Conhecimento em Bases de Dados, mais especificamente a etapa de Mineração de Dados, a fim de executar a tarefa de extração de regras de classificação em bases de dados, utilizando como método uma meta-heurística baseada no procedimento *GRASP*, denominada GRASP-DM.

Com o objetivo de construir um classificador com a máxima precisão preditiva, a metodologia adotada, que foi aplicada a sete bases de dados distintas, fez uso das diversas etapas do processo *KDD*. Na etapa de pré-processamento de dados, tanto as variáveis qualitativas quanto as quantitativas foram codificadas de maneira a corresponder a uma ou mais coordenadas binárias para o vetor de entrada. Este pré-processamento fez-se necessário, pois a meta-heurística GRASP-DM foi desenvolvida para utilizar somente coordenadas binárias na entrada.

Tal processo de codificação dos atributos, conforme pode ser observado nas bases de dados analisadas no capítulo 4, impõe, quando se trata de variável quantitativa, um aumento na cardinalidade do vetor de entrada de dados. Por outro lado, a operação destas variáveis binárias e a leitura das regras que estas irão fornecer tornam-se mais simples. Assim, pode-se afirmar que o pré-processamento adotado no presente trabalho é adequado para atender as condições de aplicação da meta-heurística proposta.

A utilização do procedimento *GRASP* como fundamento para elaboração da meta-heurística proposta atendeu as expectativas quanto à geração de regras de classificação. Conforme aludido na seção 3.3, ao se variarem os parâmetros (n_A, *SupMin* e *ConfMin*), pode-se variar o número de regras geradas, caracterizando assim a grande capacidade de adaptação desta meta-heurística às mais variadas bases de dados.

Quando uma base de dados apresentar um número pequeno de classes e de atributos, os valores de *SupMin* e/ou *ConfMin* poderão ser aumentados, o que resultará num conjunto “Melhores Regras” com um número menor de regras, porém de elevada abrangência e acurácia. Por outro lado, ao se aplicar a meta-heurística GRASP-DM a uma base de dados com número elevado de classes e atributos, ou que apresente certas peculiaridades (como, por exemplo, a base de dados zoo – seção 4.3), diminuem-se as exigências quanto aos valores de *SupMin* e/ou *ConfMin*,

gerando um número maior de regras no conjunto “Melhores Regras”, tudo isso com o objetivo de classificar todas as instâncias da base em análise. Esta facilidade na alteração dos parâmetros, que apresenta como consequência a variação do número de regras extraídas, caracteriza a versatilidade da meta-heurística proposta.

A construção do classificador é estabelecida a partir do conjunto “Melhores Regras”. As regras são ordenadas segundo o valor da sua confiança (capítulo 3). Este critério atendeu ao objetivo proposto, uma vez que os classificadores apresentam elevada precisão preditiva.

A precisão preditiva (acurácia) obtida para cada classificador de cada uma das sete bases pode ser observada na Tabela 34 do capítulo 4. Nota-se que a menor acurácia obtida pela meta-heurística GRASP-DM (base de dados *pima indians diabetes*) foi de 0,76 e que em quatro das sete bases de dados a acurácia foi superior a 0,9, chegando a 0,98 para a base de dados zoo. Assim, novamente se pode afirmar que a meta-heurística proposta cumpre a finalidade de gerar regras de classificação que constituem um classificador de elevada precisão preditiva.

A comparação estabelecida no capítulo 4 – da meta-heurística GRASP-DM com a técnica de árvores de decisão – proporcionou uma visão mais especifica das precisões preditivas que cada uma das bases analisadas apresenta para cada um dos algoritmos aplicados, quer seja o da meta-heurística GRASP-DM, foco deste trabalho, quer sejam os algoritmos envolvendo a técnica de árvores de decisão a partir do *software WEKA*: BFTree, REPTree e J4.8.

Esta comparação está apresentada de forma sucinta na Tabela 34 (capítulo 4). Pode-se observar que, para a base de dados wine, a meta-heurística GRASP-DM apresenta a mesma acurácia (0,94) da dos algoritmos J4.8 e RepTree, porém este resultado é superior ao do algoritmo BFTree (0,90). Nas outras seis bases de dados, a meta-heurística GRASP-DM apresenta acurácia superior à das demais. Destacam-se os resultados obtidos junto às bases de dados médico e *balance scale weight & distance database*, que apresentam acurácias superiores em 10% e 11%, respectivamente, em relação à segunda maior acurácia das respectivas bases. Conclui-se que a meta-heurística GRASP-DM é superior, quanto à precisão preditiva para as bases de dados analisadas, quando comparada aos algoritmos de árvore de decisão aqui apresentados.

A Tabela 34 ainda apresenta o número de regras de cada algoritmo proposto. Observa-se que a meta-heurística GRASP-DM apresenta um número maior de regras em cinco das sete bases quando comparado aos demais algoritmos; porém, este número não é relevante em termos de tempo de processamento, conforme exemplificado pelo teste estabelecido a seguir, no qual se utilizou a base de dados equilíbrio, pois seu classificador apresenta o maior número de regras (213) dentre todos os testes realizados. Esta base de dados, que contém 625 padrões, foi replicada 10 vezes gerando uma base com 6.250 instâncias. Esta base replicada foi submetida ao classificador, que gastou um tempo de processamento de 0,94 segundos para classificar todas as instâncias.

Desta forma, conclui-se que um número maior de regras apresentado pela meta-heurística GRASP-DM não dificulta nem inviabiliza o seu uso em termos de classificação automática de novos padrões.

Diante dos resultados obtidos a partir dos testes executados, nota-se que a meta-heurística GRASP-DM apresenta os requisitos básicos para executar a tarefa de *Data Mining*, mais especificamente a extração de regras de classificação. Os classificadores obtidos a partir destas regras apresentam elevadas precisões preditivas cumprindo, desta forma, o objetivo deste trabalho.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho apoiou-se no processo de Descoberta de Conhecimento em Bases de Dados; porém, como se pode observar a partir das análises nas bases de dados (capítulo 4), nem todas as etapas deste processo foram cumpridas. Como as etapas de seleção e de limpeza de dados não foram executadas, sugere-se para trabalhos futuros a efetivação de todas as etapas do processo *KDD*, segundo a metodologia adotada nesta tese.

A meta-heurística elaborada baseia-se no procedimento *GRASP* padrão. Trata-se (capítulo 2) de um procedimento multi-partidas, ou seja, que não apresenta memória; logo, cada iteração se inicia sem levar em conta a solução obtida na iteração anterior. Resende e Silva (2013) apresentam o religamento de caminhos como um grande avanço no procedimento *GRASP* padrão, pois introduzem no procedimento uma memória que resulta em melhorias significativas de desempenho e de qualidade das soluções. Assim, sugere-se a adoção de religamento de

caminhos ou de outro processo de inserção de memória ao procedimento *GRASP*, a fim de que sejam alcançadas tais melhorias.

O parâmetro *α,* utilizado na meta-heurística GRASP-DM, apresenta-se estático durante todo o processamento, ou seja, manteve-se o mesmo valor desde o início até o final da construção das regras. Sugere-se a realização de testes com este parâmetro “sendo trabalhado” de forma dinâmica. Ao iniciar a construção da regra, podem ser utilizados valores próximos a “0”, tornando o algoritmo mais aleatório e à medida que cada antecedente vai sendo acrescido à regra em construção, o valor de *α* poderia ir se aproximando de “1”, tornando a escolha dos próximos antecedentes mais gulosa.

O critério de parada adotado para a meta-heurística GRASP-DM foi o número de iterações. Testes realizados durante a sua aplicação mostraram que, a partir de certo número de iterações, poucas regras são acrescidas ao conjunto “Melhores Regras”. Este baixo aproveitamento de regras a partir deste número de iterações (que varia bastante de acordo com a base de dados) ocorre pelo fato de que as novas regras geradas, tanto na fase de construção quanto na fase de busca local, já compõem o conjunto “Melhores Regras”. Sugere-se a realização de testes nos quais seja adotado como critério de parada o baixo incremento de regras ao conjunto “Melhores Regras”.

Neste trabalho comparou-se a meta-heurística GRASP-DM com algoritmos de árvore de decisão. Sugere-se estabelecer comparações com outras meta-heurísticas cujo objetivo seja extração de regras.

As sugestões ora apresentadas – que buscam aumentar ainda mais a precisão preditiva dos classificadores obtidos – constituem apenas algumas das possibilidades de prosseguimento desta pesquisa.

REFERÊNCIAS

AGGELIS, V. ***Association rules model of e-banking services***. 5th International Conference on Data Mining, Text Mining and their Business Aplications, 2004.

BAESENS, B.; SETIONO, R.; MUES, C.; VANTHIENEN, J. ***Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evalution***. Management Science Informs, vol. 49, n° 3, p. 312-329, 2003.

BARBALHO, H.; ROSSETI, I. C. M.; MARTINS, S. L.; PLASTINO, A. ***A Hybrid Data Mining GRASP with Path-Relinking***. XLIII SBPO, Ubatuba, SP, Ago. 2011.

CAPDEVILLE, R. M. A.; VIANNA, D. S. **Heurísticas *GRASP* para o Problema de Alocação de Pontos de Acesso em uma Rede Sem Fio em Ambiente Indoor**. Sistemas & Gestão, v. 8, n. 1, p. 86-93, 2013.

CHAOVALITWONGSE, W. A.; OLIVEIRA, C. A.; CHIARINI, B.; PARDALOS, P. M.; RESENDE, M. G. C. ***Revised GRASP with path-relinking for the linear ordering problem***. J. of Combinatorial Optimization, vol. 22, pp. 572-593, 2011.

COELHO, D. G.; OLIVEIRA, M. X.; WANNER, E. F.; SOUZA, S. R. **Uma Análise da Aplicação da Meta-heurística *GRASP* ao Problema de Corte com Dimensão Aberta Guilhotinado**. XLIII SBPO. Ubatuba – SP. Agosto 2011.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHRUSAMY, R. ***Advances in knowledge Discovery & Data Mining***.California: AAAI/MIT, 1996.

FEO, T.; RESENDE, M. ***Greedy Randomized Adaptive Search Procedures***. Journal of Global Optimization, v. 6, n. 2, p. 109133, 1995.

FESTA, P.; RESENDE, M.G.C ***Hybridizations of GRASP with path-relinking.*** Studies in Computational Intelligence, v. 434, p.135-155, 2013.

FESTA, P.; PARDALOS, P.M.; PITSOULIS, L.S.; RESENDE, M.G.C., ***GRASP with path-relinking for the weighted* MAXSAT *problem*.** Journal of Experimental algorithmics , 11, Article 2.4: 1-16, 2007.

FONSECA, E. R.; FUCHSHUBER, R.; PLASTINO, A.; MARTINS, S. L. Explorando a Metaheurística Hibrida **MDM-GRASP: Uma Metaheurística Híbrida e Adaptativa**, XLI SBPO, Porto Seguro, BA, set. 2009.

FONSECA, E. R.; FUCHSHUBER, R.; SANTOS, L. F. M.; PLASTINO, A.; MARTINS, S. L. Explorando a Metaheurística Hibrida **DM-GRASP para o Problema de Multicast Confiável**, XL SBPO, João Pessoa, PB, set. 2008.

FRANÇA, M. F. **Problemas de Programação de Tarefas em Máquinas Paralelas**. Tese de Doutorado, Universidade Estadual de Campinas, Campinas, SP, 2007.

FRANCO JÚNIOR, E. F.; OLIVEIRA, H. C. B. **Adaptação da Meta-Heurística *GRASP* na Resolução do Problema de Roteamento de Veículos com Janela de Tempo**. Revista Eletrônica Pesquisa Operacional para o Desenvolvimento v.4, n.3, p. 271-287, 2012.

FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. ***Knowledge Discovery in Databases - An Overview***. In: Knowledge Discovery in Databases, pp. 1-30, 1991.

GALVÃO, N. D.; MARIN, H. F. **Técnica de mineração de dados: uma revisão da literatura.** Acta paul. enferm. vol.22 no.5 São Paulo Sept./Oct. 2009.

GARCIA, M.N. M.; ROMÁN, I.R.; PEÑALVO, F.J.G.; BONILLA, M.T. ***An association rule mining method for estimating the impact of project management policies on software quality, development time and effort***. Expert Systems with Applications, v. 34, p. 522-529, 2008.

GÓES, A. R. T; STEINER, M. T. A. **O Processo *KDD* aplicado na extração de regras: um estudo de caso da área médica**. XLIV SBPO. Rio de Janeiro – RJ. Setembro 2012.

GOLDSCHMIDT, R.; PASSOS E. **Data mining: um guia prático, conceitos, técnicas, ferramentas, orientações e aplicações**. São Paulo: Elsevier; 2005.

GONÇALVES, E. C.; MENDES, I. M. B.; PLASTINO, A. ***Mining exceptions in databases***. 17th Australian Joint Conf. on Artificial Intelligence, LNAI 3339, Cairns, Australia, 2004.

GONÇALVES, E. C.; PLASTINO, A.. ***Mining strong associations and exceptions in the stulong data set***. ECML/P*KDD* 2004 Discovery Challenge, Pisa, Itália, 2004.

GONÇALVES, E. C., **Regras de Associação e suas Medidas de Interesse Objetivas e Subjetivas,** INFOCOMP, Journal of Computer Science, 2005.

GONÇALVES, L. B.; MARTINS, S. L. & OCHI, L. S. **Uma Heurística *GRASP* para o Problema do Caixeiro Viajante Periódico**. XXXVI SBPO. São João Del Rei – MG. Novembro 2004.

HIRSCH, M.J.; PARDALOS, P. M.; RESENDE, M. G. ***Correspondence of projected 3D points and lines using a continuous GRASP*.** International Transactions in Operational Research, vol. 18, 493-511, 2011.

KARABATAK, M.; INCE, M.C. ***An expert system for detection of breast cancer based on association rules and neural network***. Expert Systems with Applications, v. 36, p. 3465-3469, 2009.

KAZIENKO, P. ***Mining Indirect Associations Rules for Web Recommendation***. International Journal of Applied Mathematics and Computer Science, v. 19, n. 1, p. 165-186, 2009.

LAROSE, D. T. ***Discovering Knowledge in Data: An Introduction to Data Mining***. John Wiley and Sons, Inc, 2005.

LOPES, A. T.; SCHULZ, V. M. L.; MAURI, G. R. ***GRASP* com *Path Relinking* para o Problema de Alocação de Berços**. Pesquisa Operacional para o Desenvolvimento V. 3 n.3, 2011.

MATEUS, G.; RESENDE, M.; SILVA, R., ***GRASP with path-relinking for the generalized quadratic assignment problem*.** Journal of Heuristics, 17, 527-565, 2011.

MESTRIA, M.; OCHI, L. S.; MARTINS, S. L. ***GRASP* com Memória Adaptativa para o Problema do Caixeiro Viajante com Grupamentos**, XXIX ENEGEP, Salvador, BA, out. 2009.

METWALLY, A.; AGRAWAL, D.; ABBADI, A. E. ***Using Association Rules for Fraud Detection in Web Advertising Networks***. 31st VLDB Conference, p. 169-180, 2005.

OLIVEIRA, M. X.; SOUZA, M. J. F.; SOUZA,S. R.; COELHO, D. G.C; PENNA, P. H. V. **Heurística *GRASP* Aplicada ao Problema de Roteamento de Veículo com Backhauls e Frota Heterogênea Fixa**. XLIV SBPO. Rio de Janeiro – RJ. Setembro 2012.

PAVANELLI, G. **Análise do tempo de duração de processos trabalhistas utilizando redes neurais artificiais como apoio a tomada de decisões**. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, PR, 2007.

PAVANELLI, G; STEINER, M. T. A.; GÓES, A. R. T.; PAVANELLI, A. M.; COSTA, D. M. B. ***Análise Extraction of Classification Rules in Databases through Metaheuristic Procedures based on GRASP***. *Advanced Materials Research* Vols. 945-949, p 3369-3375, 2014.

PITSOULIS, L.; RESENDE, M. ***Greedy Randomized Adaptive Search Procedures***. In: P.M.PARDALOS; M.G.C.RESENDE (Ed.). Handbook of Applied Optimization. [S.l.]: Oxford University Press, p. 168-181, 2002.

PLASTINO, A.; FUCHSHUBER, R.; MARTINS, S. L.; FREITAS, A. A.; SALHI, S. ***A hybrid data mining metaheuristic for the p-median problem***. *Statistical Analysis and Data Mining*, v. 4, p. 313-335, 2011.

RANGE, M. C.; ABREU, N. M. M. & BOAVANETURA, P. O. ***GRASP* para o PQA: um Limite de Aceitação para Soluções Iniciais**. Pesquisa Operacional V. 20 n.1, 2000.

RESENDE, M.; RIBEIRO, C. ***Greedy Randomized Adaptive Search Procedures***. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). Handbook of Metaheuristics. [S.l.]: Kluwer Academic Publishers, p. 219-249, 2002.

RESENDE, M. G. C.; SILVA, R. M. A., **Meta-Heurísticas em Pesquisa Operacional**. (Ed) Omnipax, 2013.

REYNOLDS, A. P. & IGLESIA, B. ***A multi-objective GRASP for partial classification.*** Soft Comput. v.13, p. 227–243, 2009.

RIBEIRO, M. H. **Incorporando Técnicas de Mineração de Dados à Metaheurística *GRASP*.** Dissertação de Mestrado, Universidade Federal Fluminense, Niterói, RJ, 2005.

RIBEIRO, M. H.; PLASTINO, A.; MARTINS, S. L.; ***Hybridization of GRASP Metaheuristic with Data Mining Techniques***, *Journal of Mathematical Modelling and Algorithms* 5: 23–41, 2006.

RIBEIRO, M. X.; TRAINA, A. J. M.; TRAINA, C.; AZEVEDO-MARQUES, P. M. ***An Association Rule-Based Method to Support Medical Image Diagnosis With Efficiency***. IEEE Transactions on Multimedia, v. 10, n. 2, 2008.

ROCHA, W. S.; BOERES, M. C. S.; RANGEL, M. C.; FERREIRA, L. B. **Aplicação das Meta-heurísticas *GRASP*, Simulated Annealing e Algoritmos Genéticos para o Problema de Tabela-horário para Universidade.** XLIV SBPO. Rio de Janeiro – RJ. Setembro 2012.

SÁNCHEZ, D.; VILA, M.A.; CERDA, L.; SERRANO, J.M. ***Association rules applied to credit card fraud detection***. Expert Systems with Applications, v. 36, p. 3630-3640, 2009.

SEMAAN, G. S.; OCHI, L. S. **Uma heurística baseada em *GRASP* para a extração de associações em bases de dados**, XIV SPOLM, set. 2011.

SILVA, M. B.; DRUMMOND, L. M. A.; OCHI, L. S. **Proposta e avaliação de heurísticas *GRASP* para o problema da diversidade máxima**. Pesquisa Operacional V. 26 n.2, 2006.

SILVA, M. B.; DRUMMOND, L. M. A.; OCHI, L. S. **Metaheurísticas GRASP+VNS para a solução de Problemas de Otimização Combinatória**, XXXII SBPO, out. 2000.

SILVA, M. M.; SUBRAMANIAN, A.; OCHI, L. S. **Uma Heurísitca Baseada em *GRASP* e *Iterated Local Search* para o Problema da Mínima Latência**, XLIII SBPO, Ago. 2011.

STEINER, M. T. A.; SOMA, N.Y.; SHIMIZU, T.; NIEVOLA, J.C.; STEINER NETO, P. J.; **Abordagem de um problema médico por meio do processo de *KDD* com ênfase à análise exploratória dos dados**. Gest Prod. 13(2):325-37, 2006.

VALE, M. M.; MOURA, D.J.; NAAS, I. A.; OLIVEIRA S. R. M.; RODRIGUES, L. H. A. ***Data mining to estimate broiler mortality when exposed to heat wave***. Sci Agric (Piracicaba, Braz), 65(3):223-9, 2008.

VIANNA, R. C. X. F.; MORO, C. M. C. B.; MOYSÉS, S. J.; CARVALHO, D.; NIEVOLA, J. C. **Mineração de dados e características da mortalidade infantil**. Cadernos de Saúde Pública (ENSP. Impresso), v. 26, p. 535-542, 2010.

WITTEN, I. H. & FRANK, E. ***Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementations***. San Francisco: Morgan Kaufmann Publishers, 2005.

YIN, P. Y.; WANG, T. Y. ***A GRASP-VNS algorithm for optimal wind-turbine placement in wind farms***. Renewable Energy 48, 489-498, 2012.

ZEFERINO, G.; AMORIM, F. M. S.; F. FILHO, A. M. F. **Algoritmos Multi-Start , *GRASP* e *ILS* Aplicados ao Problema de P-Medianas**. XLIII SBPO. Ubatuba – SP. Agosto 2011.

ZVIETCOVICH, W. G.; CARDOSO, E. M.; MANSO, J. C. G. ***Optimal allocation of meters for monitoring voltage sags and swells using the GRASP-VNS optimisation algorithm***. In: Innovative Smart Grid Technologies Latin America (ISGT LA), IEEE PES Conference On. IEEE, 2013. p. 1-5, 2013.

APENDICE - RESULTADOS OBTIDOS PARA AS BASES DE DADOS: ÍRIS, MÉDICO, *PIMA INDIANS DIABETES* E *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

1. BASE DE DADOS IRIS

Esta base de dados é muito usual na literatura de problemas de reconhecimento de padrões. Trata-se de conjunto composto de 150 amostras de plantas íris, extraída do *Machine Learning Repositttory* (*http://archive.ics.uci.edu/ml/datasets/Iris*). Cada amostra pertence a um dos três tipos de classes: íris setosa, íris versicolor e íris virgínica.

Cada uma das classes é composta por 50 instâncias, e cada indivíduo (planta) é descrito por quatro características quantitativas: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala.

1.1 PRÉ-PROCESSAMENTO DOS DADOS DA BASE DE DADOS ÍRIS

Cada um dos quatro atributos numéricos que correspondem as características da planta foram divididos em três intervalos. O critério desta divisão foi o de se manter a mesma cardinalidade dentro de cada faixa.

Cada intevalo corresponde a uma entrada binária para a meta-heurística GRASP-DM, assim, quando a planta pertence a um intervalo de determinado atributo apresentará valor “1” na coordenada correspondente do vetor de entrada de dados e, quando não pertence apresentará o valor “0”. Desta forma, cada indivíduo será caracterizado por 12 coordenadas binárias. A Tabela A1 a seguir mostra a codificação dos atributos previsores.

TABELA A1 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE ÍRIS

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A1 - Comprimento da sépala	1	4,3 <= A1<= 5,3	46
	2	5,4 <= A1<= 6,2	53
	3	6,3 <= A1<= 7,9	51

continua

TABELA A1 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE ÍRIS

conclusão

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A2 - Largura da sépala	4	2,2 <= A2 <= 2,8	47
	5	2,9 <= A2 <= 3,1	48
	6	3,2 <= A2 <= 4,4	55
A3 -Comprimento da pétala	7	1 <= A3 <= 1,9	50
	8	2 <= A3 <= 4,8	49
	9	4,9 <= A3 <= 6,9	51
A4 -Largura da pétala	10	0,1 <= A4 <= 0,6	50
	11	0,7 <= A4 <= 1,7	54
	12	1,8 <= A4 <= 2,5	46

Como a base é composta de 150 instâncias, a matriz de dados apresenta dimensão da ordem (150x12).

1.2 APLICAÇÃO DA META-HEURÍSTICA GRASP-DM PARA A BASE DE DADOS ÍRIS

Testes preliminares serviram para determinar o número máximo de antecedentes igual a três.

Os parâmetros utilizados para aplicação da meta-heurística nesta base de dados foram: critério de parada, 100 iterações; alfa igual a 0,5 (α = 0,5), confiança mínima igual a 0,5 (*ConfMin* = 0,5); e o suporte mínimo igual a 0,05 (*SupMin* = 0,05).

A Tabela A2 a seguir apresenta um dos classificadores obtido no processo de validação cruzada da meta-heurística GRASP-DM, aplicado ao grupo de treinamento.

TABELA A2 – CLASSIFICADOR IRIS APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (0,1 <= Largura da pétala <= 0,6) ENTÃO (Iris-Setosa)	45	0	90
2	SE (0,7 <= Largura da pétala <= 1,7) ENTÃO (Iris-Versicolor)	44	5	41

continua

TABELA A2 – CLASSIFICADOR IRIS APLICADO AO GRUPO DE TREINAMENTO

Conclusão

	REGRA	Class correta	Class Errada	Restam
3	SE (4,9 <= Comprimento da pétala <= 6,9) E (1,8 <= Largura da pétala <= 2,5) ENTÃO (Iris-Virgínica)	40	0	1
4	SE (2 <= Comprimento da pétala <= 4,8) ENTÃO (Iris-Versicolor)	1	0	1

A matriz de confusão apresentada na Tabela A3 a seguir apresenta os dados relevantes acerca do classificador apresentado anteriormente.

TABELA A3 – MATRIZ DE CONFUSÃO DA BASE ÍRIS APLICADO AO GRUPO DE TREINAMENTO

Classe	Íris Setosa	Íris Versicolor	Iris Virgínica	Precisão	
				Classe	Classificador
Íris Setosa	45	0	0	45/45	
Íris Versicolor	0	45	0	45/45	130/135
Iris Virgínica	0	5	40	40/45	

A partir da matriz de confusão apresentada na Tabela A3 acima nota-se que o classificador apresenta uma precisão preditiva de 96,3%, para o grupo de treinamento.

O conjunto de teste deste "*fold*" também foi submetido ao classificador apresentado na Tabela A2 acima e as precisões deste conjunto podem ser observadas na matriz de confusão conforme Tabela A4 a seguir.

TABELA A4 – MATRIZ DE CONFUSÃO DA BASE ÍRIS APLICADO AO GRUPO DE TESTE

Classe	Íris Setosa	Íris Versicolor	Iris Virgínica	Precisão	
				Classe	Classificador
Íris Setosa	5	0	0	5/5	
Íris Versicolor	0	5	0	5/5	15/15
Iris Virgínica	0	0	5	5/5	

Para este grupo de teste o classificador apresenta uma precisão preditiva de 100%, como pode ser observado na Tabela A4 acima.

Ao agrupar os 10 grupos de testes do processo de validação (*k fold*; *k* = 10) tem-se a base de dados completa apresentada como teste. A Tabela A5 a seguir apresenta os dados estatísticos acerca da acurácia da classificação das 150 amostras desta base quando estas compunham os grupos de teste.

TABELA A5 – PRECISÃO PREDITIVA DA BASE IRIS

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	96,7%	0,012	0,963
Teste	96,7%	0,105	1

A partir dos dados apresentados na Tabela A5 nota-se que a meta-heurística GRASP-DM apresenta, para esta base de dados, erro global muito pequeno, caracterizando então uma elevada eficiência na tarefa de classificação.

1.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELA META-HEURÍSTICA GRASP-DM COM A TÉCNICA DE ÁRVORES DE DECISÃO PARA A BASE DE DADOS ÍRIS

A precisão preditiva da meta-heurística GRASP-DM junto a base de dados íris, foi submetida a comparação com outros três algoritmos (BFTree, REPTree e J4.8), todos de árvore de decisão. Cabe ressaltar que foi utilizado o mesmo processo de validação (*k-fold*, com *k* = 10) tanto na meta-heurística GRASP-DM quanto nos algoritmos de árvore de decisão. A Tabela A6 a seguir apresenta o número de amostras desta base classificadas correta e incorretamente para todos os algoritmos.

TABELA A6 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS ÍRIS

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	95%	94%	96%	97%
Instâncias classificadas incorretamente	5%	6%	4%	3%

Como pode-se observar a partir da Tabela A6, a meta-heurística GRASP-DM apresentou também, para esta base de dados, resultados superiores aos obtidos pelos algoritmos de árvore de decisão.

A Tabela A7 a seguir apresenta comparações entre a meta-heurística GRASP-DM e os algoritmos de árvore de decisão, em relação as precisões preditivas das três classes e do classificador.

TABELA A7 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE ÍRIS

Algoritmos	Precisão			
	Íris Setosa	Íris Versicolor	Iris Virgínica	Classificador
BFTree	100%	94%	90%	95%
RepTree	100%	92%	90%	94%
J4.8	98%	94%	96%	96%
GRASP-DM	100%	100%	90%	97%

A partir da Tabela A7 verifica-se que a meta-heurística GRASP-DM apresentou, para esta base de dados, resultados iguais ou superiores em duas das três classes. Verifica-se ainda que o classificador proposto neste trabalho também apresentou resultados superiores aos demais algoritmos. A seguir apresenta-se o Gráfico A1, elaborado a partir das precisões das três classes que compõem esta base de dados.

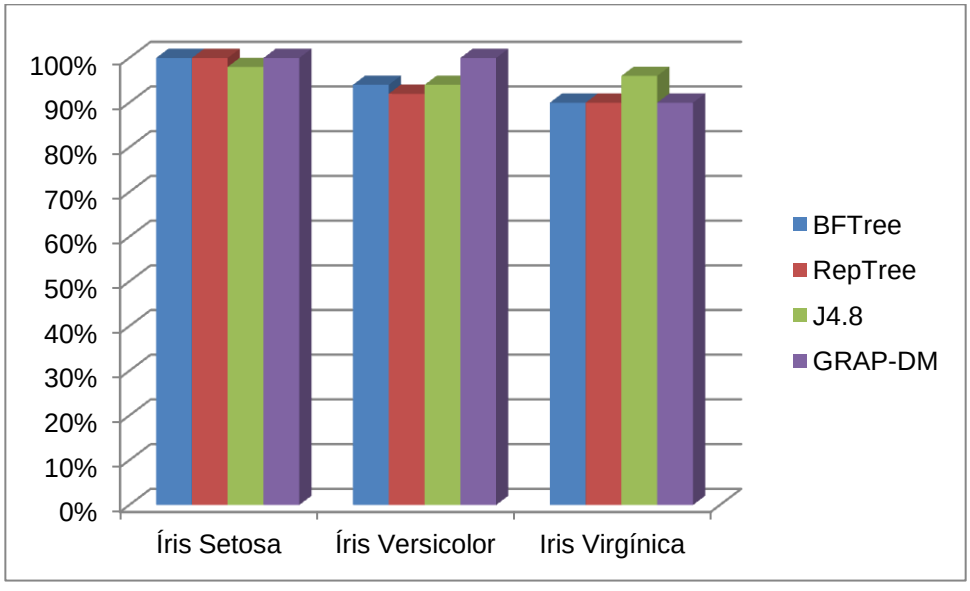

GRÁFICO A1 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS ÍRIS

2 BASE DE DADOS MÉDICO

Esta base de dados é composta de 118 registros históricos de um problema médico, apresentado por Steiner *et al*. (2006). Das 118 instâncias (pacientes), 35 possuem comprovadamente obstrução no duto biliar devido a câncer e 83, devido a cálculo. Cada instância constitui-se de 14 atributos classificatórios, a saber: idade, sexo, bilirrubina total, bilirrubina direta, bilirrubina indireta, fosfatases alcalinas, SGOT, SGPT, tempo de atividade da protrombina, albumina, amilase, creatinina, leucócitos e volume globular.

2.1 PRÉ-PROCESSAMENTO DOS DADOS DA BASE DE DADOS MÉDICO

Nesta base de dados, somente o atributo previsor sexo é do tipo nominal. Os demais atributos desta base são representados por variáveis do tipo quantitativa, e neste trabalho cada uma delas foi dividida em três intervalos, buscando manter a mesma cardinalidade em cada um deles, como se pode observar na Tabela A8.

TABELA A8 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE MÈDICO

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A1: Idade	1	A1 <= 43	40
	2	43 < A1 <= 60	41
	3	A1 > 60	37
A2: Sexo	4	A2 = 1	47
	5	A2 = 0	71
A3: Bilirrubina total	6	A3 <= 6	39
	7	6 < A3 <= 14,9	41
	8	A3 > 14,9	38
A4: Bilirrubina indireta	9	A4 <= 1,9	38
	10	1,9 < A4<= 5,93	43
	11	A4 > 5,93	37
A5: Fosfatases alcalinas	12	A5 <= 33	40
	13	33 < A5<= 86	38
	14	A5 > 86	40

continua

TABELA A8 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE MÈDICO

conclusão

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A6: SGOT	15	A6 <= 50	39
	16	50 < A6<= 89	39
	17	A6 > 89	40
A7: SGPT	18	A7 <= 148,83	38
	19	148,83 < A7<= 248,6	39
	20	A7 > 248,6	41
A8: Tempo de atividade da protrombina	21	A8 <= 82	40
	22	82 < A8 <= 144	37
	23	A8 > 144	41
A9: Albumina	24	A9 <= 13	48
	25	13 < A9 <= 14	41
	26	A9 > 14	29
A10: Amilase	27	A10 <= 2,96	37
	28	2,96 < A10 <= 3,1	40
	29	A10 > 3,1	41
A11: Creatinina	30	A11 <= 0,7	38
	31	0,7 < A11<= 0,9	40
	32	A11 > 0,9	40
A12: Leucócitos	33	A12 <= 7,7	38
	34	7,7 < A12<= 9,8	39
	35	A12 > 9,8	41
A13: Volume globular	36	A13 <= 36,8	38
	37	36,8 < A13 <= 40,5	41
	38	A13 > 40,5	39
A14: Bilirrubina direta	39	A14 <= 3,21	38
	40	3,21 < A13 <= 9,1	40
	41	A13 > 9,1	40

Conforme apresentado na Tabela A8 os atributo quantitativos foram divididos em três faixas, cada qual representada por três coordenadas binárias. Quando o atributo corresponder ao intervalo da faixa, a coordenada correspondente rebebe valor “1”, caso contrário recebe valor “0”. Por exemplo, se a idade em uma determinada instância for igual a 50 (A1 = 50) a coordenada 1 e a coordenada 3 apresentarão valor “0” enquanto que a coordenada 2 apresentará valor “1”.

2.2 APLICAÇÃO DA META-HEURÍSTICA GRASP-DM PARA A BASE DE DADOS MÉDICO

Com o objetivo de extrair regras que classifiquem corretamente as instâncias da base de dados foi aplicada a meta-heurística GRASP-DM. Testes preliminares serviram de base para estabelecer o número máximo de antecedentes da regra como quatro.

Os parâmetros utilizados para aplicação da meta-heurística nesta base de dados foram: critério de parada igual a 100 iterações; alfa igual a 0,5 (α = 0,5), a confiança mínima, igual a 0,5 (*ConfMin* = 0,5); e o suporte mínimo igual a 0,05 (*SupMin* = 0,05).

A Tabela A9 a seguir apresenta um dos classificadores obtido no processo de validação cruzada (*k fold*) aplicado ao seu respectivo grupo de treinamento, composto de 106 instâncias.

TABELA A9 – CLASSIFICADOR MÉDICO APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (Bilirrubina indireta <= 1,9) E (Bilirrubina total <= 6) E (36,8 <Volume Globular <= 40,5) ENTÃO (CÁLCULO)	16	0	90
2	SE (Bilirrubina indireta <= 1,9) E (Bilirrubina total <= 6) E (Fosfatases alcalinas <= 33) ENTÃO (CÁLCULO)	7	0	83
3	SE (1,9 < Bilirrubina indireta <= 5,93) E (Idade <= 43) ENTÃO (CÁLCULO)	15	0	68
4	SE (Tempo de atividade da protrombina > 144) E (Bilirrubina total <= 6) ENTÃO (CÁLCULO)	4	0	64
5	SE (82 < Tempo de atividade da protrombina <= 144) E (Bilirrubina indireta >5,93) ENTÃO (CÂNCER)	9	0	55

continua

TABELA A9 – CLASSIFICADOR MÉDICO APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRA	Class correta	Class Errada	Restam
6	SE (Bilirrubina indireta <= 1,9) E (Bilirrubina total <= 6) E (0,7 < Creatinina <= 0,9) ENTÃO (CÁLCULO)	4	0	51
7	SE (SGOT > 89) E (SGPT > 248,6) E (7,7 < Leocócitos <= 9,8) ENTÃO (CÂNCER)	4	0	47
8	SE (Volume Globular <= 36,8) E (Bilirrubina total > 14,9) E (82 < Tempo de atividade da protrombina <= 144) ENTÃO (CÂNCER)	1	0	46
9	SE (7,7 < Leocócitos <= 9,8) E (SGOT > 89) E (Sexo = Feminino) ENTÃO (CÂNCER)	1	0	45
10	SE (Idade <= 43) E (Fosfatases alcalinas > 86) E 13 < Albumina <= 14) ENTÃO (CÁLCULO)	2	0	43
11	SE (Tempo de atividade da protrombina > 144) E (SGOT <= 50) E (Fosfatases alcalinas <= 33) ENTÃO (CÁLCULO)	3	0	40
12	SE (6 < Bilirrubina total <= 14,9) E (Sexo = Masculino) E (SGPT <= 148,83) ENTÃO (CÁLCULO)	5	0	35
13	SE (6 < Bilirrubina total <= 14,9) E (Fosfatases alcalinas > 86) E (Sexo = Masculino) ENTÃO (CÁLCULO)	3	0	32
14	SE (Bilirrubina total > 14,9) E (Bilirrubina indireta >5,93) E (0,7 < Creatinina <= 0,9) E (Volume Globular <= 36,8) ENTÃO (CÂNCER)	1	0	31
15	SE (Volume Globular <= 36,8) E (Bilirrubina indireta >5,93) E (33 < Fosfatases alcalinas <=86) E (SGOT > 89) ENTÃO (CÂNCER)	2	0	29

continua

TABELA A9 – CLASSIFICADOR MÉDICO APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRA	Class correta	Class Errada	Restam
16	SE (Bilirrubina total <= 6) ENTÃO (CÁLCULO)	2	1	26
17	SE (6 < Bilirrubina total <= 14,9) E (Sexo = Masculino) ENTÃO (CÁLCULO)	1	1	24
18	SE (Idade <= 43) E 13 < Albumina <= 14) ENTÃO (CÁLCULO)	1	0	23
19	SE (Sexo = Masculino) E (Volume Globular > 40,5) ENTÃO (CÁLCULO)	1	1	21
20	SE (1,9 < Bilirrubina indireta <= 5,93) E (Tempo de atividade da protrombina > 144) ENTÃO (CÁLCULO)	2	0	19
21	SE (1,9 < Bilirrubina indireta <= 5,93) E (50 < SGOT <= 89) ENTÃO (CÁLCULO)	2	1	16
22	SE (Albumina <= 13) E (SGPT <= 148,83) ENTÃO (CÁLCULO)	1	0	15
23	SE (Tempo de atividade da protrombina > 144) E (Leocócitos > 9,8) ENTÃO (CÁLCULO)	0	1	14
24	SE (Tempo de atividade da protrombina > 144) E (Idade <= 43) ENTÃO (CÁLCULO)	2	0	12
25	SE (Albumina <= 13) E (Volume Globular > 40,5) ENTÃO (CÁLCULO)	1	0	11

continua

TABELA A9 – CLASSIFICADOR MÉDICO APLICADO AO GRUPO DE TREINAMENTO

conclusão

	REGRA	Class correta	Class Errada	Restam
26	SE (Sexo = Masculino) E(Fosfatases alcalinas > 86) ENTÃO (CÁLCULO)	0	2	9
27	SE (Idade > 60) E (Bilirrubina total > 14,9) ENTÃO (CÂNCER)	1	0	8
28	SE (Bilirrubina indireta >5,93) E (Volume Globular <= 36,8) ENTÃO (CÂNCER)	1	0	7
29	SE (33 < Fosfatases alcalinas <=86) E (Bilirrubina indireta >5,93) ENTÃO (CÂNCER)	1	0	6
30	SE (Bilirrubina total > 14,9) E (SGPT > 248,6) ENTÃO (CÂNCER)	1	1	4
31	SE (Sexo = Feminino) E (SGOT > 89) ENTÃO (CÂNCER)	2	1	1
32	SE (Sexo = Feminino) E (Bilirrubina total > 14,9) ENTÃO (CÂNCER)	1	0	0

A partir do classificador apresentado na Tabela A9 pode-se montar a matriz de confusão.

TABELA A10 – MATRIZ DE CONFUSÃO DA BASE MÉDICO APLICADO AO GRUPO DE TREINAMENTO

Classe	Câncer	Cálculo	Precisão	
			Classe	Classificador
Câncer	25	7	25/32	97/106
Cálculo	2	72	72/74	

A partir da matriz de confusão apresentada na Tabela A10 observa-se que o classificador apresenta uma precisão preditiva de 91,5%, para este grupo de treinamento.

Quando submetido ao classificador apresentado na Tabela A9, o conjunto de teste deste "*fold*" apresenta os resultados conforme matriz de confusão apresentada a seguir.

TABELA A11 – MATRIZ DE CONFUSÃO DA BASE MÉDICO APLICADO AO GRUPO DE TESTE

Classe	Câncer	Cálculo	Precisão	
			Classe	Classificador
Câncer	3	0	3/3	11/12
Cálculo	1	8	8/9	

A partir da matriz de confusão apresentada na Tabela A11 acima observa-se que este classificador apresenta uma precisão preditiva de 91,7%, para o grupo de teste.

Os dados relevantes acerca do processo de validação cruzada são apresentados na Tabela A12 a seguir.

TABELA A12 – PRECISÃO PREDITIVA DA BASE MÉDICO

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	91,6%	0,01	0,915
Teste	84,5%	0,094	0,917

2.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELA META-HEURÍSTICA GRASP-DM COM A TÉCNICA DE ÁRVORES DE DECISÃO PARA A BASE DE DADOS MÉDICO

Foram comparados os desempenhos da técnica de árvores de decisão – algoritmos BFTree, REPTree e J4.8 - com a meta-heurística GRASP-DM, conforme a Tabela A13 a seguir.

TABELA A13 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS MÉDICO

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	75%	77%	73%	85%
Instâncias classificadas incorretamente	25%	23%	27%	15%

Como pode-se observar a partir da Tabela A13, a meta-heurística GRASP-DM apresentou melhores resultados quando comparados aos algoritmos de árvore de decisão.

A Tabela A14 a seguir apresenta as comparações entre as precisões em cada uma das classes e também do classificador de cada uma das aplicações estabelecidas.

TABELA A14 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE MÉDICO

Algoritmos	Precisão		
	Classe 0 (Câncer)	Classe 1 (Cálculo)	Classificador
BFTree	51.4%	84,3%	74,5%
RepTree	71,4%	79,5%	77,1%
J4.8	45,7%	84,3%	72,9%
GRASP-DM	82,8%	85,5%	84,5%

A partir da Tabela A14 foi elaborado o Gráfico A2 de comparação entre as precisões apresentadas dentro de cada classe.

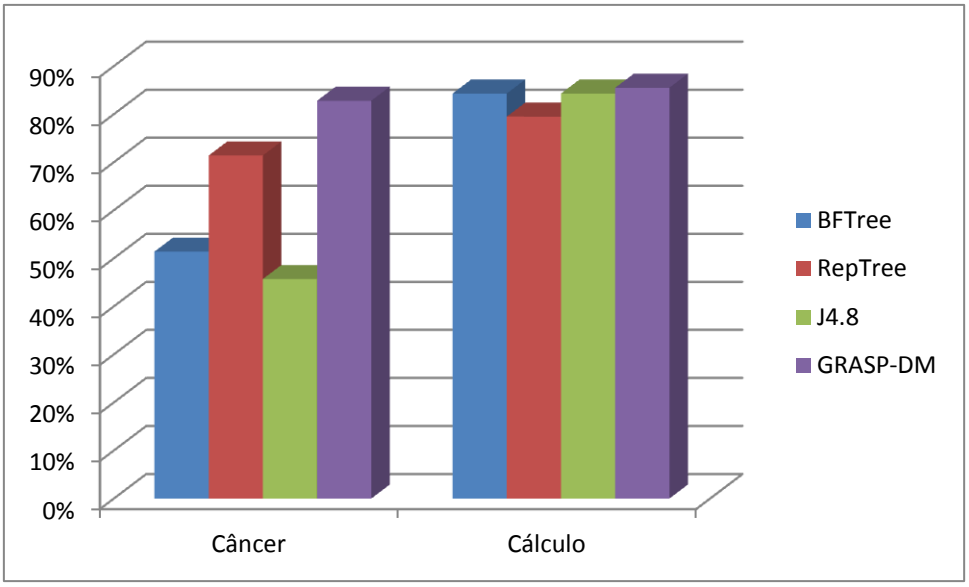

GRÁFICO A2 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS MÉDICO

3. BASE DE DADOS *PIMA INDIANS DIABETES*

Trata-se de uma base de dados constituída de 768 instâncias que investiga se a paciente (somente pacientes do sexo feminino) apresenta, de acordo com a Organização Mundial de Saúde, sinais de diabetes.

A análise envolve oito atributos classificatórios numéricos a saber: número de vezes grávida; concentração de glucose no plasma a 2 horas num teste de tolerância oral à glicose; pressão arterial diastólica (mmHg); dobras cutâneas triciptal (mm); insulina no soro (MUU/ ml); índice de massa corporal (peso em kg/(altura em m)^2); Função Diabetes e idade (anos). Cada instância pertence a uma das seguintes classes: “1” (interpretado como “teste positivo para diabetes”) ou “0” (interpretado como “teste negativo para diabetes”).

A base de dados foi extraída do *Machine Learning Repositttory* (*https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes*), e apresenta 500 padrões da classe “0” e 268 da classe “1”.

3.1 PRÉ-PROCESSAMENTO DOS DADOS DA BASE DE DADOS *PIMA INDIANS DIABETES*

Os atributos foram categorizados e codificados de maneira que cada um possa ser representado por uma ou mais coordenadas binárias a fim de compor o vetor de entrada de dados para a meta-heurística GRASP-DM. A Tabela A15 a seguir apresenta a codificação proposta para esta base de dados.

TABELA A15 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE *PIMA INDIANS DIABETES*

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A1: Número de vezes grávida	A1=1	1	111
	A1=2	2	135
	A1=3	3	103
	A1=4	4	75
	A1=5	5	68
	A1=6	6	57
	A1=7	7	50

continua

TABELA A15 – CODIFICAÇÃO DOS ATRIBUTOS PREVISORES DA BASE *PIMA INDIANS DIABETES*

conclusão

Atributo	Coordenadas	Intervalos	Número de padrões em cada intervalo
A1: Número de vezes grávida	A1=8	8	45
	A1=9	9	38
	A1=10	10	28
	A1=11	11	24
	A1=12	12	11
	A1=13	13	9
	A1=14	14	10
	A1=15	15	2
	A1=16	16	1
	A1=17	17	1
A2: Concentração de glucose	A2<= 104	18	251
	104 < A2<= 128	19	245
	A2 > 128	20	272
A3: Pressão arterial	A3<= 65	21	243
	65 < A3<= 76	22	275
	A3 > 76	23	250
A4: Dobras cutâneas	A4<= 13	24	260
	13 < A4<= 29	25	243
	A4 > 29	26	265
A5: Insulina	A5= 0	27	374
	A5 > 0	28	394
A6: Indice de massa corporal	A6<= 28	29	252
	28< A6<= 34,5	30	251
	A6 > 34,5	31	265
A7: Função Diabetes	A7<= 0,264	32	250
	0,264< A7<= 0,514	33	249
	A7 > 0,514	34	268
A8: Idade	A8<= 25	35	267
	25 < A8<= 36	36	247
	A8 > 36	37	254

Como a base é composta de 768 instâncias, a matriz de dados apresenta dimensão da ordem (768x37).

3.2 APLICAÇÃO DA META-HEURÍSTICA GRASP-DM PARA A BASE DE DADOS *PIMA INDIANS DIABETES*

Foram estabelecidos testes preliminares que definiram como sendo três o número máximo de antecedentes em cada regra. Aplicou-se então a meta-heurística GRASP-DM, adotando como parâmetros alfa igual a 0,5 e critério de parada 100 iterações. A partir das regras obtidas pela meta-heurística proposta foi construído o classificador, no qual as regras são ordenadas segundo ordem decrescente da confiança (Capítulo 3). A Tabela A16 a seguir apresenta um dos classificadores obtidos para esta base de dados.

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

	REGRAS	Class Correta	Class Errada	Ainda Restam
1	SE (Concentração de Glucose <=104) E (Idade <= 25) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	24	0	669
2	SE (Concentração de Glucose <=104) E (Insulina = 0) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	17	0	652
3	SE (13 < Dobras Cutâneas <=29) E (Pressão Arterial <=65) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	16	0	636
4	SE (Função Diabetes >= 0,264) E (Idade <= 25) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	10	0	626
5	SE (Função Diabetes >= 0,264) E (65 < Pressão Arterial <=76) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	10	0	616
6	SE (Concentração de Glucose <=104) E (Idade <= 25) E (Número de vezes grávida = 2 ENTÃO NEGATIVO	9	0	607
7	SE (Concentração de Glucose <=104) E (Insulina = 0) E (Idade <= 25) ENTÃO NEGATIVO	8	0	599

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
8	SE (Função Diabetes > 0,514) E (Idade <= 25) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	6	0	593
9	SE (13 < Dobras Cutâneas <=29) E (Insulina > 0) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	5	0	588
10	SE (Pressão Arterial > 76) E (Concentração de Glucose <=104) E (Dobras cutâneas <= 13) ENTÃO NEGATIVO	5	0	583
11	SE (Concentração de Glucose <=104) E (0,264 < Função Diabetes <= 0,514) E (Insulina > 0) ENTÃO NEGATIVO	5	0	578
12	SE (Idade <= 25) ENTÃO NEGATIVO	4	0	574
13	SE (Função Diabetes >= 0,264) E (Idade <= 25) E (Número de vezes grávida = 2) ENTÃO NEGATIVO	4	0	570
14	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	4	0	566
15	SE (65 < Pressão Arterial <=76) E (Índice de massa corporal <= 28) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	4	0	562
16	SE (0,264 < Função Diabetes <= 0,514) E (Idade <= 25) E (Número de vezes grávida = 2) ENTÃO NEGATIVO	3	0	559
17	SE (Idade <= 25) E (Concentração de Glucose <=104) E (0,264 < Função Diabetes <= 0,514) ENTÃO NEGATIVO	2	0	557

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
18	SE (Concentração de Glucose <=104) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	*2*	*0*	*555*
19	SE (0,264 < Função Diabetes <= 0,514) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	1	0	554
20	SE (0,264 < Função Diabetes <= 0,514) E (Índice de massa corporal <= 28) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	1	0	553
21	SE (28 < Índice de massa corporal <= 34,5) E (Idade <= 25) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	1	0	552
22	SE (Índice de massa corporal <= 28) E (65 < Pressão Arterial <=76) E (Número de vezes grávida = 2) ENTÃO NEGATIVO	1	0	551
23	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) E (Número de vezes grávida = 2) ENTÃO NEGATIVO	1	0	550
24	SE (65 < Pressão Arterial <=76) E (Índice de massa corporal <= 28) E (Idade <= 25) ENTÃO NEGATIVO	1	0	549
25	SE (Pressão Arterial <=65) E (Insulina > 0) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	1	0	548
26	SE (Índice de massa corporal <= 28) E (65 < Pressão Arterial <=76) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	9	1	538
27	SE (Idade <= 25) E (13 < Dobras Cutâneas <=29) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	6	1	531

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
28	SE (Concentração de Glucose <=104) E (13 < Dobras Cutâneas <=29) E (0,264 < Função Diabetes <= 0,514) ENTÃO NEGATIVO	5	1	525
29	SE (0,264 < Função Diabetes <= 0,514) E (Índice de massa corporal <= 28) E (Pressão Arterial <=65) ENTÃO NEGATIVO	4	1	520
30	SE (Índice de massa corporal <= 28) E (Insulina > 0) E (Idade <= 25) ENTÃO NEGATIVO	4	1	515
31	SE (Insulina = 0) E (Índice de massa corporal <= 28) E (Idade <= 25) ENTÃO NEGATIVO	4	1	510
32	SE (0,264 < Função Diabetes <= 0,514) E (Idade <= 25) E (13 < Dobras Cutâneas <=29) ENTÃO NEGATIVO	3	1	506
33	SE (Função Diabetes >= 0,264) E (104< Concentração de Glucose <=128) E (Idade <= 25) ENTÃO NEGATIVO	3	1	502
34	SE (13 < Dobras Cutâneas <=29) E (65 < Pressão Arterial <=76) E (Idade <= 25) NEGATIVO	3	1	498
35	SE (13 < Dobras Cutâneas <=29) E (Pressão Arterial <=65) E (0,264 < Função Diabetes <= 0,514) ENTÃO NEGATIVO	3	1	494
36	SE (Dobras Cutâneas <=13) E (Insulina = 0) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	3	1	490
37	SE (Pressão Arterial <=65) E (Concentração de Glucose <=104) E (Idade <= 25) ENTÃO NEGATIVO	3	1	486

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
38	SE (Idade <= 25) E (13 < Dobras Cutâneas <=29) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	2	1	483
39	SE (Idade <= 25) E (13 < Dobras Cutâneas <=29) E (Pressão Arterial <=65) ENTÃO NEGATIVO	2	1	480
40	SE (Idade <= 25) E (Pressão Arterial <=65) ENTÃO NEGATIVO	2	1	477
41	SE (Pressão Arterial > 76) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	2	1	474
42	SE (Pressão Arterial <=65) E (Concentração de Glucose <=104) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	2	1	471
43	SE (Pressão Arterial <=65) E (Concentração de Glucose <=104) E (Insulina = 0) ENTÃO NEGATIVO	2	1	468
44	SE (Idade <= 25) E (Concentração de Glucose <=104) E (Insulina > 0) ENTÃO NEGATIVO	1	1	466
45	SE (28 < Índice de massa corporal <= 34,5) E (13 < Dobras Cutâneas <=29) E (Idade <= 25) ENTÃO NEGATIVO	1	1	464
46	SE (Índice de massa corporal <= 28) E (Insulina > 0) ENTÃO NEGATIVO	1	1	462
47	SE (Insulina > 0) E (Número de vezes grávida = 2) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	1	1	460

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
48	SE (Insulina = 0) E (Idade <= 25) ENTÃO NEGATIVO	1	1	458
49	SE (Insulina = 0) E (Idade <= 25) E (13 < Dobras Cutâneas <=29) ENTÃO NEGATIVO	1	1	456
50	SE (Concentração de Glucose <=104) E (Função Diabetes >= 0,264) E (Pressão Arterial <=65) ENTÃO NEGATIVO	1	1	454
51	SE (Idade <= 25) E (Pressão Arterial <=65) E (Número de vezes grávida = 2) ENTÃO NEGATIVO	0	1	453
52	SE (28 < Índice de massa corporal <= 34,5) E (Idade <= 25) ENTÃO NEGATIVO	0	1	452
53	SE (13 < Dobras Cutâneas <=29) E (65 < Pressão Arterial <=76) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	0	1	451
54	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) E (Idade <= (13 < Dobras Cutâneas <=29)) ENTÃO NEGATIVO	62	2	387
55	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	13	2	372
56	SE (Idade <= 25) E (Pressão Arterial <=65) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	7	2	363
57	SE (0,264 < Função Diabetes <= 0,514) E (Dobras cutâneas <= 13) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	7	2	354

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
58	SE (Insulina > 0) E (Número de vezes grávida = 2) E (Idade <= 25) ENTÃO NEGATIVO	6	2	346
59	SE (Insulina = 0) E (Função Diabetes >= 0,264) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	6	2	338
60	SE (Índice de massa corporal <= 28) E (Função Diabetes >= 0,264) E (Pressão Arterial <=65) ENTÃO NEGATIVO	4	2	332
61	SE (0,264 < Função Diabetes <= 0,514) E (65 < Pressão Arterial <=76) E (Idade <= 25) ENTÃO NEGATIVO	3	2	327
62	SE (0,264 < Função Diabetes <= 0,514) E (Idade <= 25) ENTÃO NEGATIVO	2	2	323
63	SE (Insulina > 0) E (Número de vezes grávida = 2) E (Concentração de Glucose <=104) ENTÃO NEGATIVO	2	2	319
64	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) E (Pressão Arterial <=65) ENTÃO NEGATIVO	2	2	315
65	SE (Concentração de Glucose <=104) ENTÃO NEGATIVO	1	2	312
66	SE (Idade > (25< Idade > 36)) E (Índice de massa corporal >= 34,5) E (Concentração de Glucose > 128) ENTÃO POSITIVO	18	3	291
67	SE (13 < Dobras Cutâneas <=29) E (Pressão Arterial <=65) ENTÃO NEGATIVO	1	3	287

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

continuação

	REGRAS	Class Correta	Class Errada	Ainda Restam
68	SE (0,264 < Função Diabetes <= 0,514) ENTÃO NEGATIVO	14	4	269
69	SE (Insulina = 0) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	10	4	255
70	SE (Índice de massa corporal <= 28) E (Função Diabetes >= 0,264) ENTÃO NEGATIVO	6	4	245
71	SE (Concentração de Glucose <=104) E (0,264 < Função Diabetes <= 0,514) ENTÃO NEGATIVO	5	4	236
72	SE (Concentração de Glucose <=104) E (Insulina = 0) ENTÃO NEGATIVO	1	4	231
73	SE (13 < Dobras Cutâneas <=29) E (Índice de massa corporal <= 28) ENTÃO NEGATIVO	6	5	220
74	SE (13 < Dobras Cutâneas <=29) ENTÃO NEGATIVO	7	6	207
75	SE (Pressão Arterial <=65) ENTÃO NEGATIVO	7	8	192
76	SE (Índice de massa corporal >= 34,5) E (Concentração de Glucose > 128) E (Insulina = 0) ENTÃO POSITIVO	36	9	147
77	SE (Dobras Cutâneas >29) E (Concentração de Glucose > 128) ENTÃO POSITIVO	21	10	116

continua

TABELA A16 – CLASSIFICADOR *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

conclusão

	REGRAS	Class Correta	Class Errada	Ainda Restam
78	SE (Função Diabetes >= 0,264) ENTÃO NEGATIVO	14	10	92
79	SE (Índice de massa corporal >= 34,5) E (Concentração de Glucose > 128) ENTÃO POSITIVO	19	11	62
80	SE (104< Concentração de Glucose <=128) ENTÃO NEGATIVO	6	13	43
81	SE (Concentração de Glucose > 128) ENTÃO POSITIVO	27	16	0

A partir do classificador apresentado na Tabela A16 acima pode-se montar a matriz de confusão.

TABELA A17 – MATRIZ DE CONFUSÃO DA BASE *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TREINAMENTO

Classe	Posisitvo	Negativo	Precisão	
			Classe	Classificador
Positivo	121	117	121/238	527/693
Negativo	49	406	406/455	

A partir da matriz de confusão apresentada naTabela A17 acima observa-se que o classificador apresenta uma precisão preditiva de 76%, para o grupo de treinamento.

O conjunto de teste deste "*fold*" também foi submetido ao classificador apresentado na Tabela A16 e os dados relevantes deste conjunto podem ser observadas na matriz de confusão conforme Tabela A18 a seguir.

TABELA A18 – MATRIZ DE CONFUSÃO DA BASE *PIMA INDIANS DIABETES* APLICADO AO GRUPO DE TESTE

Classe	Posisitvo	Negativo	Precisão	
			Classe	Classificador
Positivo	17	13	17/30	59/75
Negativo	3	42	42/45	

A partir da matriz de confusão apresentada na Tabela A18 acima observa-se uma precisão preditiva no conjunto de teste de 78,7%.

Ao agrupar os 10 grupos de testes do processo de validação cruzada (*k fold*; $k = 10$) tem-se a base de dados completa apresentada como teste. A Tabela A19 a seguir apresenta os dados estatísticos acerca da acurácia da classificação das 768 instâncias desta base quando estas compunham os grupos de teste.

TABELA A19 – PRECISÃO PREDITIVA DA BASE *PIMA INDIANS DIABETES*

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	76,8%	0,022	0,760
Teste	76,3%	0,054	0,779

3.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELA META-HEURÍSTICA GRASP-DM COM A TÉCNICA DE ÁRVORES DE DECISÃO PARA A BASE DE DADOS *PIMA INDIANS DIABETES*

Com o objetivo de validar a meta-heurística GRASP-DM, estabeleceu-se comparações com algoritmos de árvores de decisão BFTree, REPTree e J4.8, obtidos a partir do *software WEKA*. A primeira comparação foi em relação a precisão preditiva do classificador, ou seja, quantas instâncias foram classificadas corretamente em cada um dos métodos. Os resultados são apresentados na tabela a seguir.

TABELA A 20 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS *PIMA INDIANS DIABETES*

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	74%	75%	74%	76%
Instâncias classificadas incorretamente	26%	25%	26%	24%

Como pode-se observar a partir da Tabela A20 acima, a meta-heurística GRASP-DM apresentou melhor precisão preditiva quando comparada com os demais algorítmos, ou seja, classificou corretamente um número maior de padrões quando comparada aos algoritmos de árvore de decisão.

Comparou-se também as precisões das classificações dentro de cada uma das classes e também do classificador de cada uma das aplicações estabelecidas. Os resultados apresentam-se na Tabela A21 a seguir.

TABELA A21 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE *PIMA INDIANS DIABETES*

Algoritmos	Precisão		
	Negativo	Positivo	Classificador
BFTree	87,6%	47,4%	73,6%
RepTree	84,6%	57,8%	75,3%
J4.8	81,4%	40,3%	73,8%
GRASP-DM	89,4%	51,9%	76,3%

A partir da Tabela A21, pode-se observar que a meta-heurística GRASP-DM apresentou maior precisão preditiva em relação aos demais algoritmos aqui apresentados, tanto para a classe “Negativo” como para o classificador. Em relação a classe “Positivo”, a meta-heurística GRASP-DM apresentou melhores resultados em duas das três comparações. A seguir apresenta-se o Gráfico A3, elaborado a partir das precisões preditivas das classes que compõem esta base de dados.

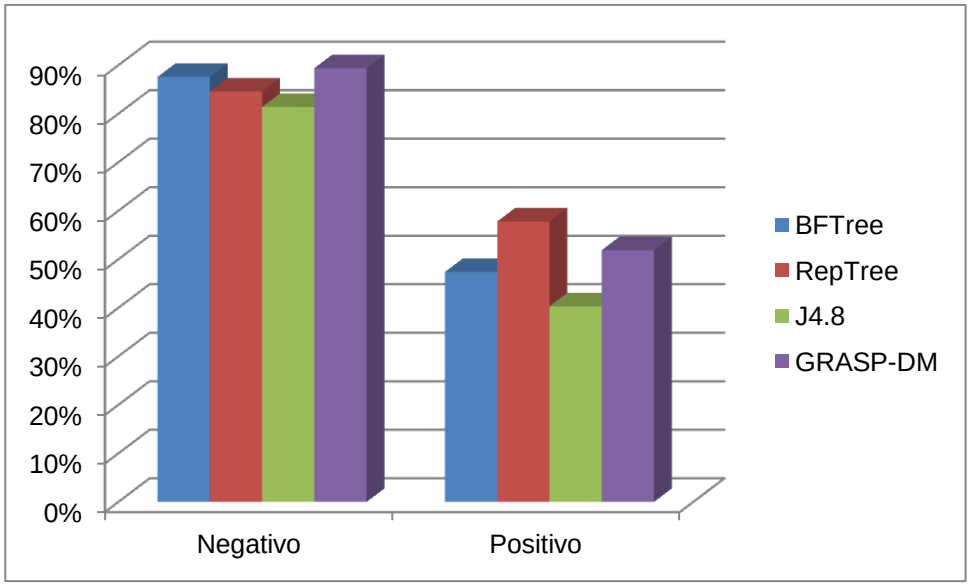

GRÁFICO A3 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS *PIMA INDIANS DIABETES*

4. BASE DE DADOS *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Trata-se de uma base de dados constituída de 625 instâncias que se classificam segundo a apresentação do ponteiro da balança em três classes distintas. A primeira classe *balanced* (B) conta com 49 amostras e caracteriza-se pelo equilíbrio do ponteiro da balança; a classe *left* (L), constituída de 288 amostras, representa os padrões em que o ponteiro da balança volta-se para a esquerda; e a terceira classe, *right* (R) é composta de 288 amostras cujo ponteiro da balança aponta para a direita.

Essa base de dados foi extraída do *Machine Learning Repositttory* (*http://archive.ics.uci.edu/ml/machine-learning-databases/balance-scale/balance-scale.names*). Cada registro apresenta quatro atributos: peso no prato esquerdo da balança, tamanho do braço esquerdo da balança, peso no prato direito da balança e tamanho do braço direito da balança. A maneira exata de encontrar a classe de um padrão é comparar o produtos do pesos no prato da balança pelos seus respectivos comprimentos dos braços. Assim, se o produto do comprimento do braço esquerdo pelo peso esquerdo for maior do que o produto do braço direito pelo peso direito, trata-se da classe L, caso contrário trata-se da classe R. No caso dos produtos de serem iguais, classe B.

4.1 PRÉ-PROCESSAMENTO DOS DADOS DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Os quatro atributos desta base são representados por variáveis do tipo quantitativa discreta, todas com valores variando de 1 a 5. Neste trabalho cada uma delas foi dividida em cinco intervalos, buscando manter a mesma cardinalidade em cada um deles, como se pode observar na Tabela A22.

TABELA A22 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Atributo	Coordenadas	Valor	Número de padrões em cada intervalo
A1 – Peso no prato esquerdo da balança	1	1	125
	2	2	125
	3	3	125

continua

TABELA A22 - CODIFICAÇÃO DOS ATRIBUTOS DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

conclusão

Atributo	Coordenadas	Valor	Número de padrões em cada intervalo
A2 – Comprimento do braço esquerdo da balança	4	4	125
	5	5	125
	6	1	125
	7	2	125
	8	3	125
	9	4	125
	10	5	125
A3 – Peso no prato direito da balança	11	1	125
	12	2	125
	13	3	125
	14	4	125
	15	5	125
A4 – Comprimento do braço direito da balança	16	1	125
	17	2	125
	18	3	125
	19	4	125
	20	5	125

Conforme apresentado na Tabela A22, cada atributo foi dividido em cinco faixas, sendo então representado por cinco coordenadas binárias.

4.2 APLICAÇÃO DA META-HEURÍSTICA GRASP-DM PARA A BASE DE DADOS *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Com o objetivo de se extraírem regras que classifiquem corretamente as instâncias da base de dados, foi aplicada a meta-heurística GRASP-DM. Testes preliminares serviram de base para estabelecer o número máximo de antecedentes da regra como três.

Os parâmetros utilizados para aplicação da meta-heurística nesta base de dados foram: o critério de parada, 100 iterações; alfa igual a 0,5 (α = 0,5); a

confiança mínima igual a 0,3 (*ConfMin* = 0,5); o suporte mínimo igual a 0,05 (*SupMin* = 0,05).

Dentre as base de dados analisadas neste trabalho, esta foi a que apresentou o classificador com maior número de regras. Desta forma, a Tabela A23 a seguir apresenta apenas uma parte das regras que compõem um dos classificadores obtidos no processo de validação cruzada (*k fold*) e aplicado ao seu respectivo grupo de treinamento.

TABELA A23 – CLASSIFICADOR *BALANCE SCALE WEIGHT & DISTANCE DATABASE* APLICADO AO GRUPO DE TREINAMENTO

	REGRA	Class correta	Class Errada	Restam
1	SE (Peso no prato esquerdo da balança = 3) E (Peso no prato direito da balança = 1) E (Comprimento do braço esquerdo da balança = 3) ENTÃO (Classe *Left*).	5	0	494
2	SE (Peso no prato esquerdo da balança = 3) E (Peso no prato direito da balança = 2) E (Comprimento do braço esquerdo da balança = 4) ENTÃO (Classe *Left*).	5	0	489
3	SE (Peso no prato esquerdo da balança = 3) E (Peso no prato direito da balança = 1) E (Comprimento do braço esquerdo da balança = 4) ENTÃO (Classe *Left*).	5	0	484
...	...	...	...	...
12	SE (Peso no prato esquerdo da balança = 1) E (Peso no prato direito da balança = 3) E (Comprimento do braço esquerdo da balança = 1) ENTÃO (Classe *Right*).	5	0	439
13	SE (Peso no prato esquerdo da balança = 1) E (Peso no prato direito da balança = 4) E (Comprimento do braço esquerdo da balança = 1) ENTÃO (Classe *Right*).	5	0	434
14	SE (Peso no prato esquerdo da balança = 1) E (Comprimento do braço direito da balança = 5) E (Comprimento do braço esquerdo da balança = 2) ENTÃO (Classe *Right*).	5	0	429
...	...	...	...	...

continua

TABELA A23 – CLASSIFICADOR *BALANCE SCALE WEIGHT & DISTANCE DATABASE* APLICADO AO GRUPO DE TREINAMENTO

conclusão

	REGRA	Class correta	Class Errada	Restam
182	SE (Comprimento do braço esquerdo da balança = 2) E (Peso no prato direito da balança = 2) ENTÃO (Classe *Balanced*).	4	5	81
183	SE (Peso no prato esquerdo da balança = 2) E (Peso no prato direito da balança = 2) ENTÃO (Classe *Balanced*).	4	1	76
184	SE (Comprimento do braço esquerdo da balança = 2) E (Comprimento do braço direito da balança = 2) ENTÃO (Classe *Balanced*).	3	7	66
...	...	...	...	...
213	SE (Peso no prato esquerdo da balança = 3) ENTÃO (Classe *Balanced*).	3	0	0

A partir do classificador apresentado na Tabela A23, pode-se montar a Matriz de confusão.

TABELA A24 – MATRIZ DE CONFUSÃO DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE* APLICADO AO GRUPO DE TREINAMENTO

Classe	*Balanced*	*Right*	*Left*	Precisão	
				Classe	Classificador
Balanced	23	11	5	23/39	
Right	11	217	2	217/230	465/499
Left	5	0	225	225/230	

A partir da matriz de confusão constante na Tabela A24, observa-se que o classificador apresenta uma precisão preditiva de 93,2% para este grupo de treinamento.

Quando submetido ao classificador explicitado na Tabela A25, o conjunto de teste deste "*fold*" exibe os resultados conforme matriz de confusão apresentada a seguir.

TABELA A25 – MATRIZ DE CONFUSÃO DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE* APLICADO AO GRUPO DE TESTE

Classe	*Balanced*	*Right*	*Left*	Precisão	
				Classe	Classificador
Balanced	2	6	2	2/10	
Right	10	48	0	48/58	101/126
Left	7	0	51	51/58	

A partir da matriz de confusão da Tabela A25 acima observa-se que este classificador apresenta uma precisão preditiva de 80,2% para o grupo de teste.

Os dados relevantes acerca do processo de validação cruzada são apresentados na Tabela A26, a seguir.

TABELA A26 – PRECISÃO PREDITIVA DA BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Conjunto	Acurácia Global	Desvio Padrão	Mediana
Treinamento	93%	0,006	0,93
Teste	90%	0,059	0,926

4.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELA META-HEURÍSTICA GRASP-DM COM A TÉCNICA DE ÁRVORES DE DECISÃO PARA A BASE DE DADOS *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Foram comparados os desempenhos da técnica de árvores de decisão – algoritmos BFTree, REPTree e J4.8 – com a meta-heurística GRASP-DM. A Tabela A27, apresenta o número de instâncias classificadas correta e incorretamente para todos os algoritmos.

TABELA A27 – CLASSIFICAÇÃO DAS INSTÂNCIAS SEGUNDO OS ALGORITMOS APLICADOS À BASE DE DADOS *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

	BFTree	REPTree	J4.8	GRASP-DM
Instâncias classificadas corretamente	79%	77%	77%	90%
Instâncias classificadas incorretamente	21%	23%	23%	10%

Como se pode observar a partir da Tabela A27, a meta-heurística GRASP-DM apresentou melhores resultados quando comparados aos algoritmos REPTree, BFTree e J4.8.

A Tabela A28, a seguir, apresenta as comparações entre as precisões em cada uma das classes e também do classificador de cada uma das aplicações estabelecidas.

TABELA A28 – COMPARATIVO DA PRECISÃO DOS ALGORITMOS APLICADOS À BASE *BALANCE SCALE WEIGHT & DISTANCE DATABASE*

Algoritmos	Precisão			
	Balanced	Right	Left	Classificador
BFTree	2%	85,8%	84,7%	78,7%
RepTree	0	85,1%	82,6%	77,3%
J4.8	0	84,7%	81,6%	76,6%
GRASP-DM	51%	95,8%	92%	90,1%

A partir da Tabela A28, foi elaborado o Gráfico A4 de comparação entre as precisões apresentadas dentro de cada classe.

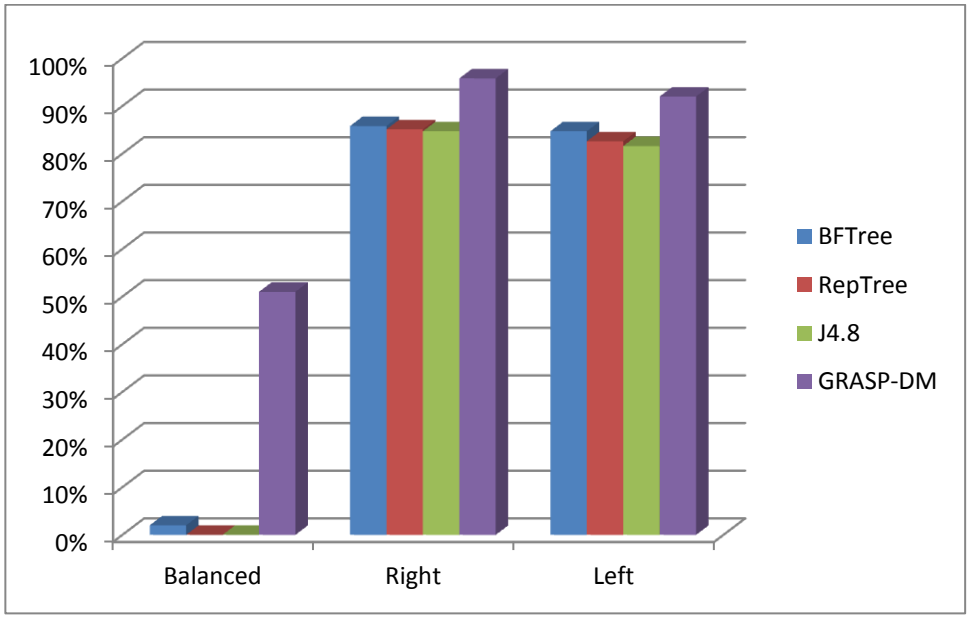

GRÁFICO A4 – COMPARATIVO ENTRE AS PRECISÕES APRESENTADAS DENTRO DE CADA CLASSE PARA BASE DE DADOS *BALANCE SCALE WEIGHT & DISTANCE DATABASE*