Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

Algoritmos genéticos para seleção de atributos em problemas de classificação de processos de negócio

Márcio Porto Basgalupp

Porto Alegre
Janeiro de 2007

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

Algoritmos genéticos para seleção de atributos em problemas de classificação de processos de negócio

Márcio Porto Basgalupp

Dissertação apresentada como requisito parcial à obtenção do grau de mestre em Ciência da Computação

Orientadora: Profª. Drª. Vera Lúcia Strube de Lima

Porto Alegre
Janeiro de 2007

Pontifícia Universidade Católica do Rio Grande do Sul

Dados Internacionais de Catalogação na Publicação (CIP)

B299a Basgalupp, Márcio Porto

Algoritmos genéticos para seleção de atributos em problemas de classificação de processos de negócio / Márcio Porto Basgalupp. – Porto Alegre, 2007.

77 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.

Orientador: Prof ª. Dr ª. Vera Lúcia Strube de Lima.

1. Informática. 2. Algoritmos Genéticos.

3. Processos de negócio. 4. Seleção de Atributos. I. Título.

CDD 005.1

Ficha Catalográfica elaborada pelo Setor de Processamento Técnico da BC-PUCRS

PUCRS

Campus Central

Av. Ipiranga, 6681 – prédio 16 – CEP 90619-900

Porto Alegre – RS – Brasil

Fone: +55 (51) 3320-3544 – Fax: +55 (51) 3320-3548

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "***Algoritimos Genéticos para Seleção de Atributos em Problemas de Classificação de Processos de Negócio***", apresentada por Márcio Porto Basgalupp, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 11/01/2007 pela Comissão Examinadora:

Vera Lúcia Strube de Lima

Profa. Dra. Vera Lúcia Strube de Lima – Orientador (a) PPGCC/PUCRS

Prof. Dr. Duncan Dubugras Alcoba Ruiz – PPGCC/PUCRS

Prof. Dr. Stanley Loh – UCP/RS

Homologada em 18/03/2008 conforme Ata No. 05/2008. pela Comissão Coordenadora.

Fernando Luis Dotti

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P. 16 – sala 106 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@inf.pucrs.br
www.pucrs.br/facin/pos

Para minha mãe, avó e minhas irmãs.

Agradecimentos

Aos professores e amigos Dr. João Baptista da Silva, Dr. João Artur de Souza e Drª. Gertrudes Aparecida Dandolini, os quais contribuíram muito para o meu crescimento profissional e pessoal.

À minha querida orientadora Profª. Drª. Karin Becker, a quem devo tudo o que aprendi durante o mestrado.

Ao Prof. Dr. Duncan Dubugras Ruiz pelo apoio em todos os momentos em que precisei.

À minha também orientadora Profª. Drª. Vera Lúcia Strube de Lima pela dedicação, atenção e disponibilidade na reta final do meu trabalho.

À empresa Hewlett-Packard pelo apoio financeiro durante todo o tempo do mestrado.

Aos colegas de mestrado, com os quais compartilhei bons momentos e que, diretamente ou indiretamente, influenciaram neste trabalho.

À minha mãe Neida Mariza Veiga Porto e ao meu pai Jorge Luiz Basgalupp (*in memoriam*), responsáveis diretos por cada uma de minhas realizações.

Aos meus demais familiares e amigos, simplesmente por existirem e fazerem parte da minha vida.

Resumo

Um processo de negócio define um conjunto de atividades junto com os seus possíveis fluxos de execução e recursos necessários. Trabalhos da área de *Business Intelligence* (BI) têm destacado o papel da mineração de dados como instrumento facilitador da análise, previsão e otimização de processos de negócio. Uma das tarefas mais utilizadas da mineração de dados é a classificação, cujo objetivo é, dado um conjunto de dados ou instâncias de treino, induzir um modelo preditivo capaz de associar a cada instância sua classe ou categoria. Espera-se que este modelo seja bem sucedido na classificação de novas instâncias. No contexto de processos de negócio, o uso da classificação tem como objetivo entender as causas de determinados comportamentos e gerar modelos de predição do comportamento e do desempenho dos processos.

Problemas práticos de classificação de padrões e descoberta de conhecimento requerem a seleção de subconjuntos de atributos preditivos para representar os padrões a serem classificados, pois a presença de atributos preditivos irrelevantes, redundantes ou em grande quantidade pode prejudicar a qualidade do modelo de classificação. Em classificação de processos de negócio, é bastante interessante a utilização de seleção de atributos, visto que a quantidade de atributos que caracterizam um processo pode ser enorme. Além dos atributos diretamente relacionados a uma instância de processo, também devem ser considerados os atributos pertencentes às atividades contidas neste processo.

Assim, este trabalho propõe a utilização de algoritmos genéticos multiobjetivos para seleção de atributos em problemas de classificação de processos de negócio. Os resultados obtidos foram considerados satisfatórios, visto que os critérios utilizados na função de *fitness*, ou seja, os critérios a serem otimizados, foram melhorados. Problemas específicos do domínio de processos de negócio foram detectados. Esses problemas surgem em virtude da presença de caminhos alternativos e ordem de execução das atividades nos fluxos de processos. Embora tais problemas não sejam tratados no presente trabalho, são apresentadas possíveis soluções a serem abordadas em trabalhos futuros.

Palavras-chave: processos de negócio, seleção de atributos, classificação e algoritmos genéticos.

Abstract

A business process defines a set of activities along with their possible execution flows and their necessary resources. Business Intelligence (BI) projects have been show the importance of the data mining techniques to analysis, prediction and optimization of business processes. One of the most important data mining's tasks is classification, which aims, by a training dataset or a set of training instances, the induction of a predictive model capable to associate each instance to its respective class or category. In the business process context, the aim of classification task is to understand the causes of certain behaviors and to generate models to predict the behavior and performance of these processes.

Practical problem in pattern classification and knowledge discovery tasks require the selection of predictive attribute sets in order to represent the patterns which will be classified. This is because the presence of irrelevant and redundant attributes may damage the quality of classification models. When leading with business processes classification, it is recommended the use of feature selection, due to the large possible amount of attributes may be necessary to characterize a process. In addition to the attributes that are directly related to the process, it must also considered other attributes related to each process activity.

Thus, this work aims the use of multiobjective genetic algorithms for feature selection upon business processes' classification problems. The obtained results were satisfactory, considering that the criteria aimed to be optimized were improved. Specific business process domain problems were detected. These problems appears due to the presence of alternative paths and execution order of the processes flow tasks. Although those problems are not considered in the present work, we presented possible solutions to be adopted in future studies.

Keywords: business processes, feature selection, pattern classification and genetic algorithm.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

CRM	*Customer Relationship Management*	25
SCM	*Supply Chain Management*	25
ERP	*Enterprise Resource Planning*	25
BPM	*Business Process Management*	25
WfMS	*Workflow Management System*	25
BI	*Business Intelligence*	25
MGAFS	*Multiobjective Genetic Algorithm for Feature Selection*	27
WfMC	*Workflow Management Coalition*	29
MDL	*Minimum Description Langth*	34
AG	Algoritmos Genéticos	36
SVM	*Support Vector Machine*	41
LDA	*Linear Discriminant Analysis*	41
SFS	*Sequential Forward Selection*	42
SBE	*Sequential Backward Elimination*	42
BPI	*Business Process Intelligence*	44
PME	*Process Mining Engine*	44
iBOM	*Intelligence Business Operation Management*	44
APM	*Abstract Process Monitor*	45
FAPE	*Factor Analysis and Prediction Engine*	45
CFS	*Correlation-Based Feature Selection*	45

Sumário

1 Introdução

Um processo de negócio define um conjunto de atividades com os seus possíveis fluxos de execução e recursos necessários. Como o mercado está cada vez mais competitivo, as organizações buscam proporcionar serviços ou produtos de forma rápida, com custos baixos e com maior qualidade e segurança. Nesse cenário, os primeiros esforços se focaram na automatização dos processos de negócio, viabilizando um melhor controle e gestão dos mesmos. Para suprir essas necessidades, muitas soluções foram desenvolvidas, tais como *Customer Relationship Management* (CRM), *Supply Chain Management* (SCM), *Enterprise Resource Planning* (ERP), *Business Process Management* (BPM) e *Workflow Management System* (WfMS).

Em especial, o WfMS vai ao encontro das estratégias de redesenho e otimização dos processos de negócio através da automação dos seus fluxos de trabalho, tornando-os mais ágeis, seguros, confiáveis e proporcionando um diferencial para a organização. Além disso, possibilita realizar um registro das execuções dos processos de negócio, também denominado *log* de execução. A análise desse registro de execução através de recursos analíticos apropriados pode permitir que as organizações aprimorem seus processos de negócio e aumentem a qualidade dos serviços prestados a seus parceiros.

Entretanto, as empresas não estão mais interessadas em apenas acompanhar e controlar o andamento de seus processos, mas sim em medi-los, associar valores à execução de cada instância de processo, atividade e recurso (tanto humano quanto computacional) e analisar seu desempenho. Além disso, elas buscam soluções mais pró-ativas, que sejam capazes de prever eventuais anomalias de execução, que monitorem e disparem avisos de alerta ou notificações para as pessoas responsáveis e identifiquem eventuais áreas de melhoria no processo. De modo geral, o interesse das organizações passou a ser a análise, monitoração e previsão de seus processos de negócio [5–7]. Neste contexto, surgiu a idéia de utilizar técnicas de descoberta de conhecimento para realizar tal análise, permitindo o controle e acompanhamento dos processos bem como a otimização e predição dos mesmos [6].

A Figura 1 ilustra um exemplo de um modelo de processo de negócio. Podem ser observadas características como ciclo, paralelismo e caminhos alternativos para execução das atividades. As empresas possuem processos normalmente muito mais complexos, gerando um número muito grande de caminhos possíveis de serem percorridos.

Trabalhos da área de Inteligência de Negócio (*Business Intelligence* ou BI) têm enfatizado a necessidade dos gestores de ter um melhor controle de suas operações e de como as mesmas estão relacionadas aos objetivos do negócio [5, 6]. Trabalhos como [6, 7] destacam o papel das técnicas de mineração de dados para a análise, previsão e otimização de processos de negócio.

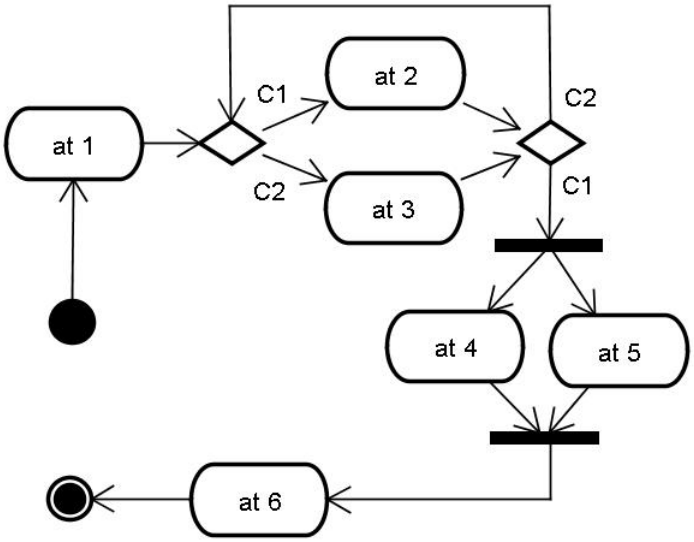

Figura 1: Diagrama de atividades que representa o exemplo de um modelo de processo de negócio.

Uma das técnicas de mineração de dados mais usadas é a classificação [8], cujo objetivo geral é construir um modelo conciso de distribuição de um atributo denominado classe em função dos demais atributos, designados preditivos. Este modelo é construído a partir de um conjunto de dados de treinamento (conjunto de treino) e o resultado é utilizado para atribuir valores ao atributo classe de dados onde somente os atributos preditivos são conhecidos.

No contexto de processos de negócio, o objetivo da classificação é entender as causas de determinados comportamentos e gerar modelos para análises descritivas e preditivas do comportamento e do desempenho destes processos. Para isso, é feito um mapeamento dos comportamentos em um problema de classificação, onde instâncias de processos são os objetos a serem classificados e a classe é o comportamento da instância [6].

Existem várias maneiras de representar um modelo de classificação: regras de classificação, árvores de decisão, fórmulas matemáticas, redes neurais artificiais, redes bayesianas e algoritmos genéticos, dentre outras. O modelo a ser usado é escolhido de acordo com o interesse da área de aplicação. A utilização das árvores de decisão como modelo de classificação de processos de negócio é bastante intuitiva, visto que as mesmas representam regras, facilitando a interpretação dos resultados. Cabe reafirmar que, se o objetivo é entender as causas de certos comportamentos dos processos, é essencial que o modelo de classificação tenha uma boa interpretabilidade.

De acordo com [3], problemas práticos de classificação de padrões e de descoberta de conhecimento requerem a seleção de subconjuntos de atributos preditivos para representar os padrões a serem classificados, pois a presença de atributos preditivos irrelevantes, redundantes ou em grande quantidade, além de não contribuir, pode prejudicar a qualidade do modelo de classificação. Em classificação de processos de negócio, é bastante interessante a utilização de seleção de atributos, visto que a quantidade de atributos que caracterizam um processo pode ser enorme.

Além dos atributos diretamente relacionados a um instância de processo, também deve-se considerar os atributos pertencentes às atividades contidas neste processo. Dependendo da disposição das atividades no fluxo do processo, algumas delas podem ser executadas inúmeras vezes (ciclos), aumentando ainda mais o número de atributos, e outras podem nem mesmo ser executadas (caminhos alternativos), gerando uma grande quantidade de dados faltantes. Esses fatores implicam baixa qualidade dos modelos resultantes, alto custo computacional e, principalmente, dificuldade de interpretação dos modelos.

Dependendo do tipo de aplicação, a seleção de atributos pode envolver diferentes critérios a ser otimizados. A facilidade de combinar vários critérios como objetivo de otimização, dentre outras vantagens, torna a utilização da técnica de algoritmos genéticos bastante atrativa na construção de métodos de seleção de atributos em problemas de classificação [3, 4, 9, 10]. Isto permite que, para um determinado problema, um conjunto de soluções ótimas seja apresentado sem privilegiar um ou outro objetivo, mas sim deixando a cargo do usuário escolher a solução que mais se adapte às suas necessidades.

Neste contexto, formula-se a seguinte questão de pesquisa:

- A utilização de um método de seleção de atributos baseado em algoritmos genéticos multiobjetivos pode melhorar a qualidade do classificador, otimizando critérios importantes, e ajudar na descoberta de problemas específicos do domínio de classificação de processos de negócio?

Para responder à questão supracitada, o objetivo deste trabalho é propor a utilização de algoritmos genéticos multiobjetivos para seleção de atributos em problemas de classificação de processos de negócio. No intuito de qualificar o modelo de classificação obtido, espera-se encontrar problemas específicos à classificação de processos e, se possível no método de seleção de atributos, propor formas para solucioná-los.

O restante desta dissertação é organizado como segue: o Capítulo 2 apresenta a base teórica que sustenta esta pesquisa, descrevendo os principais conceitos envolvidos em processos de negócio (*workflow*), algoritmos genéticos, seleção de atributos e classificação. No Capítulo 3 são apresentados trabalhos relacionados tanto à utilização de algoritmos genéticos para seleção de atributos quanto à importância de um modelo de classificação qualificado para descoberta de conhecimento em processos de negócio. O Capítulo 4 apresenta a descrição e validação do protótipo MGAFS (*Multiobjective Genetic Algorithm for Feature Selection*). No Capítulo 5 são apresentados os experimentos realizados para mineração de processos de negócio. Por fim, no Capítulo 6, encontra-se as considerações finais.

2 Fundamentação Teórica

Neste capítulo são apresentados os principais conceitos necessários para o entendimento do trabalho. Primeiro, o leitor tem um contato com o embasamento teórico sobre automação de processos de negócio (*workflow*). Após, é apresentada a técnica de classificação. Em seguida, encontram-se os principais conceitos sobre seleção de atributos. Por fim, é trazido o embasamento sobre algoritmos genéticos.

2.1 Processos de Negócio

A WfMC, *Workflow Management Coalition* [11], é uma organização sem fins lucrativos, composta por empresas e pesquisadores, que tem por objetivo oportunizar a exploração de tecnologia de *workflow* através do desenvolvimento de padrões e terminologia comuns nesta área. De acordo com [11], *workflow* é definido como a automação de processos de negócios onde documentos, informações ou tarefas são passadas de um participante para outro de acordo com um conjunto de regras para atingir ou contribuir para um objetivo de negócio.

Para gerenciar computacionalmente esses processos, é definido um Sistema de Gestão de *Workflow* (WfMS). O WfMS é um sistema que define, cria e gerencia a execução de *workflows* através do uso de software, executando um ou mais *workflow engines*, os quais são capazes de interpretar a definição do processo, interagir com os participantes do *workflow* e, quando requisitado, invocar o uso de recursos e aplicações [11]. Esses sistemas visam atender às necessidades de organizações no que diz respeito, principalmente, à automação de seus processos de negócios. Nesse contexto, um processo de negócio é composto por um conjunto de atividades relacionadas entre si que, coletivamente, atendem a um objetivo da empresa. A Figura 2 ilustra os principais conceitos envolvidos na representação de processos de negócio por *workflows*.

Um processo, nesse contexto, é a representação de um processo de negócio em um formato que aceite suporte automatizado. A definição de um processo consiste em uma rede de atividades e seus relacionamentos, critérios para indicar o início e o término do processo e informações sobre cada atividade, tais como participantes, dados e aplicativos associados. Os processos também podem ser decompostos em subprocessos.

Atividade é a descrição de uma unidade de trabalho que representa uma etapa dentro de um processo. Uma atividade requer recursos humanos (atividades manuais) ou automatizados (atividades automáticas) para sua realização. Sempre que um recurso humano é necessário, uma

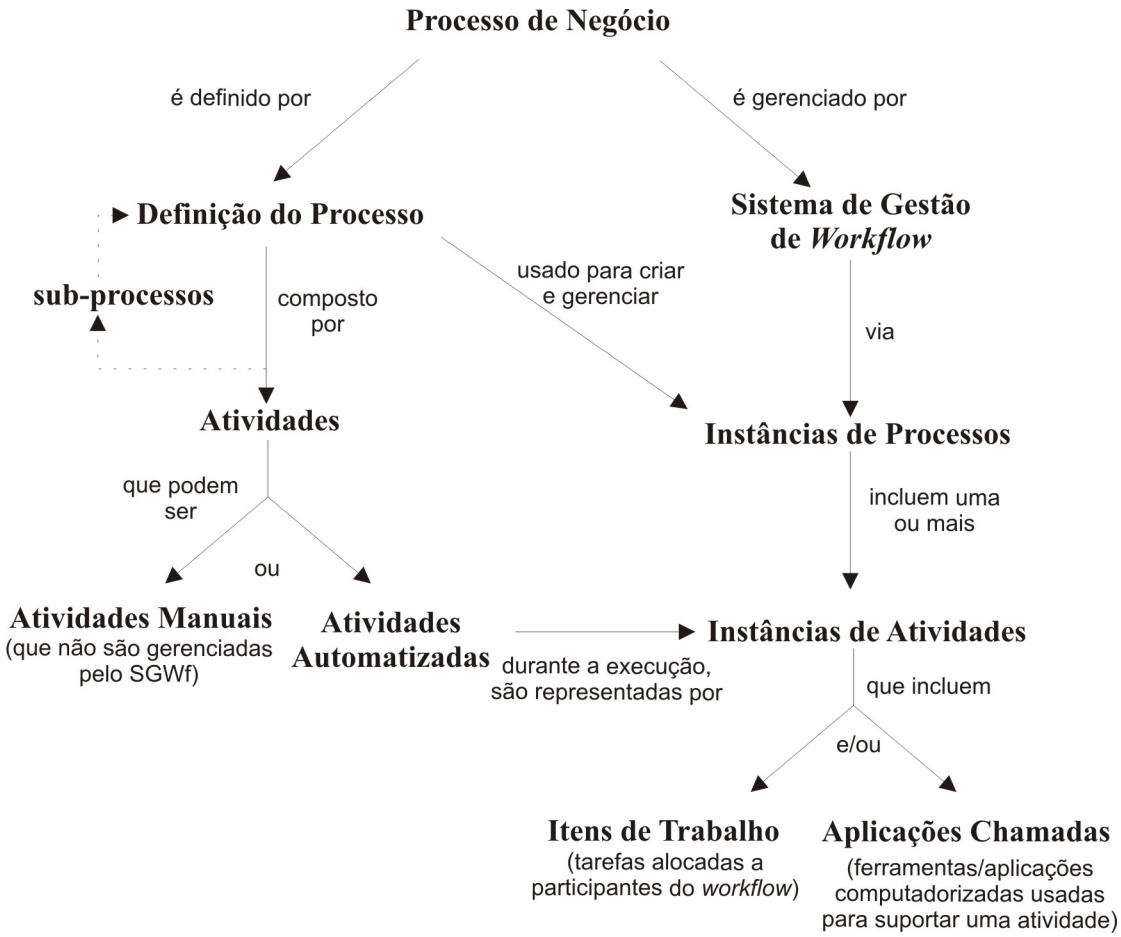

Figura 2: Relacionamentos entre os ítens da terminologia básica.

atividade é alocada a um participante do *workflow*, denominado ator do sistema.

Uma instância é uma ocorrência de um processo, ou de uma atividade em um processo, incluindo seus dados associados. Cada instância pode representar uma linha diferente de execução (por exemplo, em um processo que inclui atividades em paralelo), pode ser controlada independentemente pelo WfMS, possui estados internos próprios e tem sua identidade externamente visível.

Participante do *workflow* ou ator é quem executa o trabalho representado por uma instância de atividade. Atores podem ser pessoas ou processos automatizados. Ambos os tipos de atores são tratados em um mesmo nível de abstração com o objetivo de modelar a interação propriamente dita. Essa natureza híbrida, ou seja, aspectos humanos e computacionais tratados de forma homogênea, é característica de sistemas de *workflow*.

De acordo com a análise dos conceitos envolvidos em automação de *workflow*, é possível selecionar os principais dados que devem ser armazenados nos registros de execução dos processos, com vista à utilização desses registros para a descoberta de conhecimento. Informações mais detalhadas sobre quais dados podem ser armazenados são encontradas em [12], onde é apresentado o modelo dos dados registrados pelo Oracle *Workflow*.

2.2 Classificação

O objetivo geral da classificação é construir um modelo conciso de distribuição de um atributo denominado classe em função dos demais atributos, designados preditivos. Este modelo é construído a partir de um conjunto de dados de registros de treino (conjunto de treino) e o resultado é utilizado para designar valores ao atributo classe de registros onde somente os atributos preditivos são conhecidos.

O modelo de classificação também é conhecido como classificador. Um classificador é útil para os seguintes propósitos:

- ***análise descritiva***: o classificador pode servir para explicar as características dos objetos de diferentes classes. Por exemplo, pode ser usado para informar as características dos clientes que são bons pagadores. Essas informações são obtidas apenas com a interpretação do modelo de classificação gerado através de exemplos conhecidos.

- ***análise preditiva***: o classificador é utilizado para classificar objetos que não foram utilizados na construção do modelo. Por exemplo, pode ser usado para predizer se um novo cliente vai ser um bom ou mau pagador.

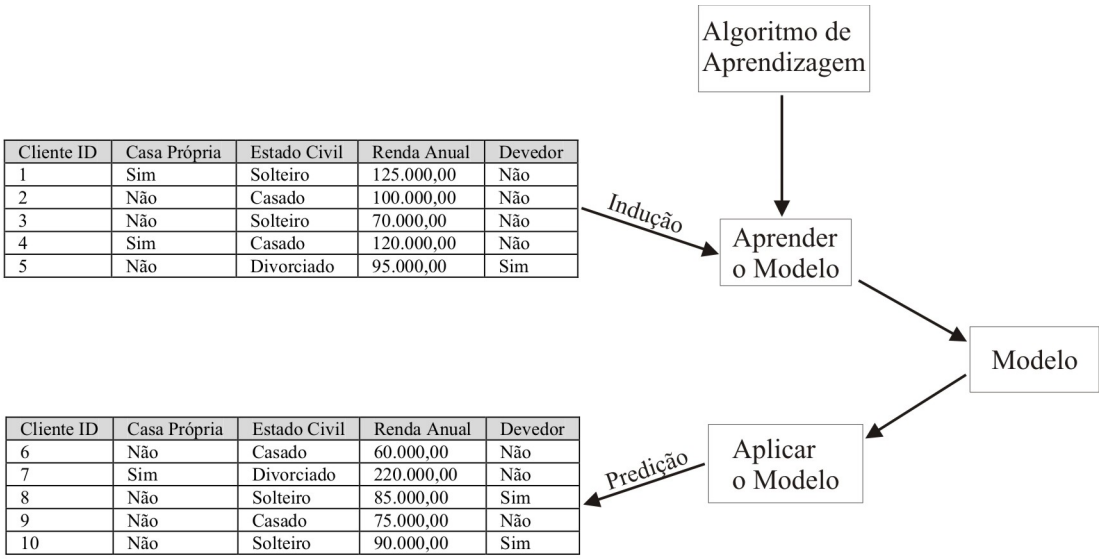

Figura 3: Abordagem geral para construir um modelo de classificação.

A técnica de classificação é uma abordagem sistemática para construir classificadores a partir de um conjunto de treino. Existem diversas técnicas de representar um classificador, tais como árvores de decisão, redes neurais artificiais, algoritmos genéticos, redes bayesianas, fórmulas matemáticas, dentre outras. Cada técnica emprega um algoritmo de aprendizagem para gerar o modelo que melhor se ajuste aos dados que compõem o conjunto de treino. Entretanto,

o objetivo principal de um algoritmo de aprendizagem é construir um modelo que seja capaz de generalizar, ou seja, predizer, com um alta taxa de acerto, as classes dos objetos que não foram utilizados na construção do modelo.

A Figura 3 ilustra a abordagem geral do processo de classificação. Primeiro, um conjunto de treino, o qual é composto de registros onde o atributo classe é conhecido, deve ser fornecido. Após, o modelo construído a partir do conjunto de treino é aplicado ao conjunto de teste, onde os valores para o atributo classe não são conhecidos.

A seguir, é feita uma sumarização sobre a técnica de árvores de decisão, a qual é utilizada na presente pesquisa em virtude da sua facilidade de interpretação. Após, são destacados os problemas de *overfitting* e *underfitting*, os quais estão relacionados à complexidade dessa técnica.

2.2.1 Árvores de Decisão

Uma árvore de decisão é uma estrutura que pode ser utilizada para, através de simples regras de decisão, dividir sucessivamente uma grande coleção de registros em conjuntos menores. A cada divisão realizada, os dados são separados de acordo com características em comum até chegar a pontos indivisíveis, que representam as classes. Cada nodo da árvore representa um teste a ser realizado e as arestas definem um caminho para cada resposta desses testes. O nodo raiz é aquele que não possui nenhuma aresta de entrada, havendo zero ou mais arestas na saída. Os nodos internos possuem exatamente uma aresta de entrada e duas ou mais arestas de saída. Os nodos folhas da árvore são os pontos indivisíveis, os quais representam as classes. A Figura 4 ilustra um exemplo de árvore de decisão para classificar clientes como devedores ou não.

Para classificar um registro utilizando uma árvore de decisão, basta começar pelo nodo raiz da árvore, onde é aplicado o primeiro teste com o atributo referente a este nodo. O processo se repete até ser encontrado um nodo folha, o qual representa o valor associado pela árvore ao atributo classe do registro em questão.

Conforme [8], as principais características dos algoritmos de aprendizagem para construção das árvores de decisão são:

- Computacionalmente econômicos, tanto para a construção como para a utilização do modelo.
- Árvores de decisão, especialmente árvores pequenas, são relativamente fáceis de interpretar. A acurácia das árvores também é comparável à de outras técnicas de classificação.
- A presença de atributos redundantes não é prejudicial à acurácia das árvores de decisão. Quando um atributo é escolhido para dividir os dados, os redundantes a ele não serão usados posteriormente. Entretanto, atributos irrelevantes podem ser usados acidentalmente durante a construção da árvore, resultando em árvores mais complexas e menos precisas.

- Uma sub-árvore pode ser replicada múltiplas vezes na árvore de decisão, tornando-a mais complexa que o necessário e talvez mais difícil de interpretar.

- As árvores de decisão possuem representação limitada para modelar relacionamentos complexos entre atributos contínuos. Isto porque os nodos teste apresentam apenas um atributo como condição de teste, ou seja, é testado apenas um atributo por vez.

- Estudos têm mostrado que a escolha da medida de impureza tem pouco efeito no desempenho dos algoritmos de indução de árvores de decisão. Isto porque muitas dessas medidas são muito consistentes umas com as outras. De fato, a estratégia usada para podar as árvores tem um maior impacto na árvore final do que a escolha da medida de impureza.

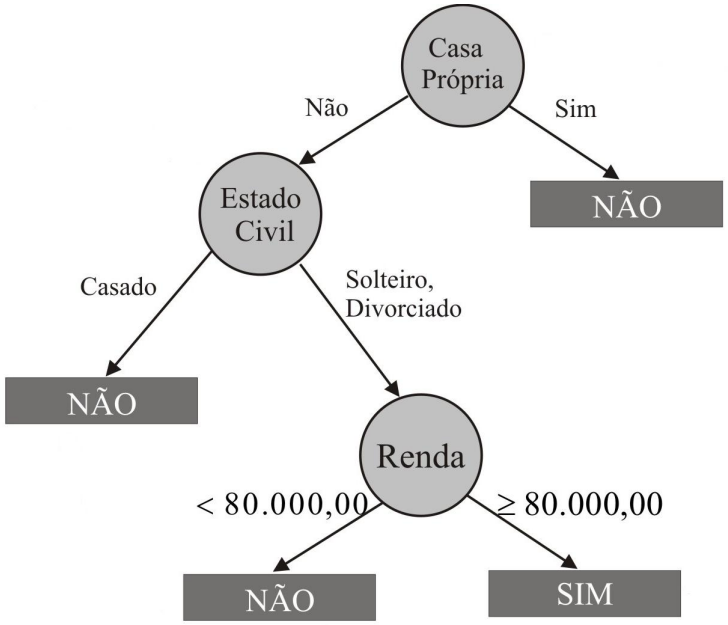

Figura 4: Uma árvore de decisão para o problema de classificação de clientes devedores.

2.2.2 *Overfitting* e *Underfitting*

Um modelo que representa muito bem os dados de treino pode apresentar um erro de generalização maior que um modelo que tenha um maior erro para o conjunto de treino. Quando isso ocorre, tem-se o problema conhecido como *overfitting*, ou seja, aprende demasiadamente o conjunto de treino mas não é capaz de generalizar. Quando um modelo apresenta valores altos para ambos os erros (treino e generalização), tem-se o problema de *underfitting*.

Overfitting e *underfitting* são dois problemas que estão relacionados à complexidade do modelo de classificação. Geralmente, o problema de *underfitting* surge na tentativa de resolução

do *overfitting*. De acordo com [8], as principais ocorrências de *overfitting* são causadas por presença de ruídos nos dados ou por falta de exemplos representativos (diversidade dos dados).

A principal estratégia utilizada para eliminar o problema de *overfitting* é a poda das árvores de decisão. Esta poda pode ocorrer após a construção do modelo, caracterizando a *postpruning*, ou dinamicamente durante a construção do mesmo, a *prepruning*. Para a realização das podas, são utilizadas medidas que incorporam a complexidade do modelo. As principais medidas são *Occam's razor*, *Pessimistic Error Estimate*, e *Minimum Description Length* (MDL). Maiores detalhes sobre estas medidas, bem como sobre poda das árvores de decisão, são encontrados em [8].

2.3 Seleção de Atributos

Em problemas de classificação, dificilmente são conhecidos a priori os atributos preditivos relevantes para a determinação do atributo classe (saída ou alvo). Conforme [13], os atributos são caracterizados da seguinte forma:

- ***Relevantes:*** são aqueles que têm influência na determinação da classe e sua contribuição não pode ser suprida por outro atributo;
- ***Irrelevantes:*** são aqueles que não têm influência alguma na classe, ou seja, seus valores são considerados aleatórios para diferentes valores do atributo classe;
- ***Redundantes:*** uma redundância existe sempre que um atributo pode ser calculado em função de outro.

Para um problema de classificação, muitos dos atributos preditivos podem ser irrelevantes ou redundantes para o atributo classe, sendo desejável que estes não sejam utilizados. De acordo com [14], a seleção de atributos é definida como um processo que escolhe um subconjunto ótimo de atributos de acordo com um dado critério. A escolha desse critério pode ser influenciada pelo propósito da seleção de atributos. Dentre as vantagens da seleção de atributos em problemas de classificação estão:

- Melhoria da qualidade dos dados disponíveis;
- Aumento do desempenho nas execuções dos algoritmos de classificação;
- Melhoria da interpretabilidade dos classificadores.

A Figura 5 ilustra as etapas que compõem o processo padrão de seleção de atributos [14]. O critério de parada pode ser em função das etapas de geração ou de avaliação.

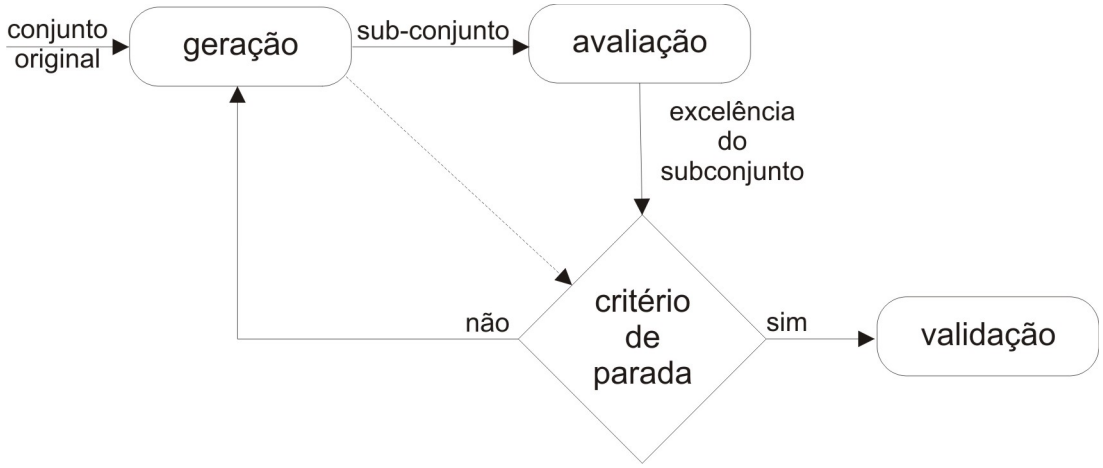

Figura 5: Processo de seleção de atributos com validação. Fonte: [1].

2.3.1 Geração

O procedimento de geração é baseado em algum tipo de método de busca. Basicamente, ele gera os subconjuntos de atributos a serem avaliados. De acordo com [1], as formas para geração dos candidatos a solução são classificadas como: completa, heurística ou aleatória.

No caso da geração completa, é realizada uma busca completa no espaço de soluções possíveis da solução ótima, de acordo com uma função de avaliação. Esse tipo de geração garante sempre a solução ótima, porém o processo pode se tornar impraticável devido ao custo computacional envolvido, visto que a complexidade é de ordem O(2^n).

A geração heurística é bastante simples de ser implementada e é muito rápida na produção dos resultados, pois sua complexidade não é tão grande (de ordem O(n^2), ou mesmo menos). Entretanto, não é garantida uma solução ótima, embora se possa chegar bem perto dela.

O procedimento de geração aleatória é relativamente novo em comparação aos outros supracitados. O espaço de busca é de ordem O(2^n), mas geralmente gera um número de subconjuntos menor que 2^n por ser restrito por um número máximo de iterações. A geração aleatória também não garante a solução ótima, mas converge para ela.

2.3.2 Avaliação e Validação

A avaliação mede a excelência de cada subconjunto candidato, ou seja, é uma função que avalia o quão bom é cada subconjunto de atributos. Quando algum algoritmo de indução de classificador é utilizado na função na avaliação, o método de seleção de atributos é *wrapper*. Caso contrário, o método é do tipo *filter*. De acordo com [1], as funções de avaliação estão divididas em cinco categorias: distância, informação, dependência, consistência e taxa de erro do classificador.

A categoria interessante para esta pesquisa é a taxa de erro do classificador, a qual caracteriza o método de seleção de atributos como *wrapper*. Geralmente, a acurácia do classificador é utilizada com função de avaliação. Pela necessidade da indução de um classificador para cada candidato solução, o processo como um todo pode se tornar bastante lento.

O processo de seleção de atributos se conclui com o procedimento de validação. Na realidade, a validação não faz parte do processo de seleção de atributos em si mas, na prática, o processo deve ser validado. A validação deve ser independente dos processos específicos de seleção de atributos, principalmente para poder compará-los.

2.4 Algoritmos Genéticos

A seguir são apresentados os conceitos básicos sobre o princípio de funcionamento de um algoritmo genético padrão. Após, é descrita uma forma de tornar multiobjetivo um algoritmo genético padrão. Conforme [15], esta forma é bastante intuitiva e é a mais indicada para resolver o problema proposto neste trabalho.

2.4.1 Funcionamento básico

Algoritmos genéticos (AG) são algoritmos de busca baseados nos mecanismos de seleção natural e genética. Eles são capazes de evoluir soluções de um problema do mundo real, sendo considerados uma abordagem muito atrativa na busca de soluções sub-ótimas em problemas de otimização. Ainda que sejam aleatórios, os algoritmos genéticos não seguem por caminhos simplesmente aleatórios, pois utilizam informações históricas (soluções anteriores) para especular os novos pontos de busca no espaço de soluções [16, 17].

Um AG trabalha iterativamente sobre um conjunto de possíveis soluções para um determinado problema. Esse conjunto constitui uma população e cada possível solução corresponde a um indivíduo, também conhecido como cromossomo, dessa população.

O processo de evolução é dirigido pela função de *fitness*, a qual atribui o valor de aptidão para cada indivíduo da população. Essa função varia de acordo com o problema a ser solucionado e deve ser muito bem estudada. As etapas envolvidas na execução de um algoritmo genético padrão podem ser visualizadas na Figura 6.

O processo iterativo de um AG inicia com a criação da população inicial de cromossomos, a primeira geração. A seguir, estes cromossomos são avaliados pela função de *fitness*. Após, caso um critério de parada não seja satisfeito, começa a evolução da população, iniciando com a etapa de seleção, onde são escolhidos os indivíduos que formarão a próxima geração da população. Quanto maior for o valor de *fitness* de um indivíduo, maior é a probabilidade de ele ser selecionado. Logo depois, aos pares, os indivíduos selecionados começam a fase de reprodução, onde

Figura 6: Diagrama de atividades utilizado para representar um algoritmo genético básico.

são realizadas operações genéticas de *crossover* para a geração de seus descendentes (filhos). Após serem gerados todos os filhos, uma política de substituição é utilizada, determinando de que maneira será formada a próxima geração da população. Formada a nova população, cada indivíduo pode passar, ainda, por um processo chamado de mutação. A mutação ocorre de acordo com uma probabilidade (geralmente muito baixa), provocando alterações aleatórias nos genes dos indivíduos.

Na etapa de evolução da população, são trabalhadas questões como a forma de seleção (ex: roleta, torneio e *ranking*) dos indivíduos que formarão as próximas gerações e os operadores genéticos utilizados, como *crossover* (ex: simples, multiponto e uniforme) e mutação (ex: aleatória).

A forma de codificação do problema em questão em cromossomos e a escolha de uma adequada função de *fitness* são os fatores chaves do sucesso, ou não, da aplicação da técnica. A escolha apropriada da técnica de geração de população inicial, do método de seleção e da forma de reprodução também pode influenciar bastante no resultado final sendo, muitas vezes, realizados testes exaustivos com diferentes escolhas, visando encontrar a melhor escolha para cada problema. Como o algoritmo genético é não determinístico, a validação do seu resultado como solução do problema não deve ser realizada apenas com uma única execução do mesmo, mas sim com várias execuções para então se calcular uma média, tornando o resultado mais confiável.

2.4.2 Algoritmos Genéticos Multiobjetivos

Existem diversas abordagens de algoritmos genéticos multiobjetivos [17]. Quando se tem algum conhecimento sobre o domínio do problema a ser tratado, como é o caso do presente trabalho, a forma mais indicada e intuitiva é agregar diferentes critérios em uma única função de *fitness*, onde são atribuídos pesos a cada um destes critérios. Uma função exemplo desta abordagem para um problema de maximização é dada em (1).

$$max \sum_{i=1}^{k} w_i f_i(x) \tag{2.1}$$

onde $w_i \geq 0$ é o coeficiente peso representando a importância do i-ésimo critério ($1 \leq i \leq k$) para o problema tratado.

Esta abordagem é bastante simples, fácil de implementar e eficiente, pois não requer nenhuma mudança no mecanismo básico do algoritmo genético. Um problema bastante óbvio nesta abordagem é a dificuldade de gerar os pesos ideais para cada critério. Este problema pode ser tratado com a ajuda de um especialista do domínio ou com uma boa experimentação buscando encontrar estes pesos empiricamente.

3 Trabalhos Relacionados

Não foram localizados, na literatura, trabalhos que tratem de métodos de seleção de atributos especificamente para classificação de processos de negócio. Nesse sentido, este capítulo é dividido em duas partes. A primeira apresenta artigos que utilizam algoritmos genéticos na seleção de atributos para problemas genéricos de classificação. A segunda parte descreve trabalhos que destacam a importância da seleção de atributos para classificação de processos de negócio.

3.1 Seleção de atributos com algoritmos genéticos

Existem diversos trabalhos na literatura que fazem uso da técnica de algoritmos genéticos para seleção de atributos, tanto para problemas de classificação quanto para outros tipos de aplicação como, por exemplo, agrupamento (*clustering*).

Os algoritmos genéticos podem ser utilizados tanto na condição *filter* como na condição *wrapper* de seleção de atributos. Quando, para o cálculo da função de *fitness* do AG, é necessária a indução de um classificador, este AG é um método *wrapper* de seleção de atributos. Caso contrário, o AG é considerado um método *filter* de seleção de atributos. Na grande maioria dos trabalhos encontrados na literatura, os AGs são utilizados como *wrapper*.

A seguir, são analisados alguns artigos que utilizam algoritmos genéticos para seleção de atributos em problemas de classificação.

3.1.1 Yang & Honavar

Yang & Honavar em [3] propõem um método *wrapper* para seleção de atributos utilizando algoritmos genéticos. Os autores desejam verificar a eficácia desta abordagem para classificação de padrões utilizando redes neurais artificiais. As redes neurais são construídas pelo algoritmo DistAl ([18] *apud* [3]), o qual é considerado por eles como sendo simples e rápido na aprendizagem do conjunto de dados de treino. Logo, a utilização do DistAL pode melhorar consideravelmente o desempenho da seleção de atributos quando são utilizadas as redes neurais como classificadores.

O algoritmo genético empregado possui uma configuração padrão [17]. Foi utilizado o

método de seleção *ranking* e, sobre a geração da população inicial, que é considerada uma das etapas mais importantes do algoritmo genético, nada foi manifestado.

A codificação dos indivíduos é feita através de um *string* binário de tamanho igual ao número de atributos, onde cada posição i do *string* indica se o atributo i é selecionado (valor 1) ou se não é selecionado (valor 0).

A principal contribuição de [3] é o destaque sobre a importância da utilização de uma função de *fitness* multiobjetiva, ou seja, que não leva em conta somente uma variável (acurácia de um classificador, por exemplo). Neste trabalho, a função de *fitness* é uma combinação da acurácia do classificador com um valor de custo, o qual representa o custo de se utilizar um determinado subconjunto de atributos. A idéia é que cada atributo preditivo tenha um custo a ele associado, e é desejável que os atributos com custo elevado sejam selecionados somente quando forem altamente necessários, ou seja, caso sua contribuição seja indispensável para a determinação do atributo classe. A função de *fitness* é representada pela Equação 3.1.

$$fitness(x) = accuracy(x) - \frac{cost(x)}{accuracy(x) + 1} + cost_{max} \tag{3.1}$$

onde x representa um indivíduo da população, $accuracy(x)$ é a acurácia do classificador gerado por x, $cost(x)$ é a soma dos custos do subconjunto de atributos representados por x e $cost_{max}$ é um limite superior dos custos dos candidatos solução. Neste caso, $cost_{max}$ é a soma dos custos associados com a cada um dos atributos selecionados. A determinação dos pesos dados a cada tipo de medida (neste caso, acurácia e custo) não é uma questão trivial. Na prática, deve ser utilizado o conhecimento do domínio da aplicação para isto. Na realidade é difícil expressar, em uma única função, qual é a melhor combinação de dois ou mais atributos.

Para avaliar o método proposto, foram utilizados diversos conjuntos de dados reais, os quais podem ser obtidos em [19] e [20]. Os resultados mostraram progresso em relação aos métodos Richeldi [21] e DistAL [18], além de melhorar os resultados sem a utilização de seleção de atributos. Também foi feita uma comparação com o mesmo algoritmo genético proposto no trabalho, porém utilizando somente a acurácia do classificador como função de *fitness*. O algoritmo genético utilizando a função de *fitness* composta por acurácia e custo forneceu resultados melhores em todos os requisitos, ou seja, menor número de atributos selecionados, maior acurácia do classificador e menor custo. Entretanto, essa medida de custo só pode ser utilizada quando são associados custos para cada atributo do conjunto de dados.

3.1.2 Sun *et al.*

Em [9], Sun *et al.* utilizam um algoritmo genético para seleção de atributos em um problema de classificação de imagens faciais para a determinação do sexo das pessoas, com o objetivo de reduzir a taxa de erro na classificação.

A representação dos indivíduos do algoritmo genético é a mesma do trabalho descrito na Seção anterior. Foram utilizados 4 classificadores diferentes para a avaliação dos indivíduos: bayesianos [22], redes neurais [23] *apud* [9], SVM (*Support Vector Machine*) [24] *apud* [9] e LDA (*Linear Discriminant Analysis*) [25] *apud* [9]. A utilização de uma função de *fitness* multiobjetiva também foi destacada neste trabalho onde, além do valor da acurácia obtida pelo classificador, também foi levado em consideração o número de atributos selecionados. Essa função de *fitness* é representada pela Equação 3.2.

$$fitness(x) = 10^4 accuracy(x) + 0,4Zeros(x) \tag{3.2}$$

onde $accuracy(x)$ indica a acurácia do classificador gerado pelo indivíduo x e $Zeros(x)$ representa o número de atributos não selecionados pelo indivíduo x.

A população do AG foi inicializada aleatoriamente, da seguinte maneira: foi sorteado aleatoriamente o número de 1s para cada indivíduo e, após, as posições do cromossomo foram sorteadas aleatoriamente para receber esses 1s. Essa técnica elimina a probabilidade de os indivíduos da população possuírem um número de 1s sempre em torno da metade do número total de atributos. O método de seleção utilizado foi o *cross generational selection* [26].

O melhor desempenho foi obtido usando o classificador SVM. Usando somente 8,4% do total de atributos, o SVM gerou uma taxa de erro de 4,7%, sendo que usando seleção manual de atributos foi obtida uma taxa de erro de 8,9%. A seleção manual é realizada por um especialista, o qual seleciona os atributos que considera relevantes de acordo com seu conhecimento sobre o domínio do problema. Esta superioridade sobre a seleção manual de atributos foi constatada na utilização dos quatro classificadores.

3.1.3 Cantú-Paz

Neste trabalho, Cantú-Paz apresenta um método híbrido de seleção de atributos, unindo um algoritmo genético simples e um método *filter* baseado em separabilidade de classes. Este método *filter* é utilizado para a geração da população inicial do algoritmo genético, o qual é utilizado como método *wrapper*. Por isso, o método proposto por [4], o qual é denominado *FilterGA*, é considerado um método híbrido de seleção de atributos. Entretanto, essa combinação se dá apenas seqüencialmente, pois o *filter* é usado uma única vez, na inicialização da população, e depois o algoritmo genético segue sua evolução apenas como método *wrapper*.

Segundo [4], o método *filter* usado elimina atributos redundantes e irrelevantes, reduzindo o espaço de busca dos algoritmos genéticos e, conseqüentemente, melhorando o desempenho do processo. Porém, os testes de redundância (entre atributos preditivos) e de relevância (atributos preditivos com o atributo classe) são realizados dois a dois. Mas, um atributo preditivo sozinho pode ser considerado irrelevante para o atributo classe e, quando unido a outro atributo preditivo,

pode ser relevante. Por isso é utilizado também um algoritmo genético como *wrapper*, visando resolver este problema durante a evolução da população.

O algoritmo genético foi implementado na sua forma padrão. O método de seleção foi o torneio e a função de *fitness* utilizou apenas a acurácia de um classificador bayesiano. Essa acurácia foi calculada através do método 2-*fold cross-validation*. Assim como nos dois trabalhos anteriores, os indivíduos do algoritmo genético são representados por um *string* binário de tamanho igual ao número de atributos.

Para avaliar se a abordagem híbrida apresenta vantagens sobre outros métodos, foi feita uma comparação em termos de acurácia e desempenho. Para isso, foram utilizados diversos conjuntos de dados, os quais estão disponíveis em [20] e [19]. As comparações foram feitas com os seguintes métodos: sGA (algoritmo genético padrão), SFS (*Sequential Forward Selection*), *Filter* (aquele usado no *FilterGA*), e SBE (*Sequential Backward Elimination*). Além destes, foi considerado também a indução de um classificador utilizando todos os atributos disponíveis.

Os resultados obtidos pelo método *FilterGA* foram considerados satisfatórios, pois proporcionaram um acréscimo significativo na acurácia do classificador bayesiano, além de melhorar o desempenho na classificação. Quando o critério de comparação foi o número de atributos selecionados, o método *FilterGA* só foi inferior ao método SFS. Talvez isso se explique pelo fato de não ser utilizada, na função de *fitness*, nenhuma informação sobre a quantidade de atributos selecionados.

3.1.4 Cherkauer & Shavlik

Em [10] *apud* [27], Cherkauer & Shavlik propõem o SET-Gen, um algoritmo genético para seleção de atributos que visa melhorar a compreensibilidade das árvores de decisão geradas pelo C4.5 [28], sem prejudicar a acurácia deste classificador.

Diferente dos outros trabalhos citados anteriormente, no SET-Gen cada indivíduo é representado por um vetor de tamanho fixo, onde cada gene pode conter um atributo ou estar vazio. Como não há uma posição fixa para cada atributo, pode ocorrer, após as operações realizadas durante a evolução do AG, de posições diferentes serem preenchidas pelo mesmo atributo. Por isso, o tamanho do vetor representa o número máximo de atributos que podem ser selecionados, sendo que esse número é atingido somente quando todas as posições são preenchidas com atributos diferentes. Essa codificação foi adotada por tornar mais lenta a perda da diversidade da população, pois os indivíduos podem variar bastante após as operações genéticas. Além disso, nesse esquema de codificação o número de genes independe do número de atributos presentes na classificação. Este tipo de codificação é denominado pelos autores como "genoma".

Para o cálculo da função de *fitness* de uma solução candidata, ou seja, um indivíduo x, são consideradas as seguintes medidas: número de atributos presentes na solução candidata (F), tamanho médio das árvores (S) e a média das taxas de classificação (A) geradas durante

a validação por *cross-validation*. Um maior peso é dado a A para manter o nível original de acurácia. A função de *fitness* é obtida pela Equação 3.3. Cabe salientar que não foi mencionado no artigo o porquê dos valores atribuídos como pesos, dando a entender que essas atribuições foram feitas intuitivamente.

$$fitness(x) = \frac{3}{4}A(x) + \frac{1}{4}(1 - \frac{S(x) + F(x)}{2}) \tag{3.3}$$

O AG implementado utiliza os operadores tradicionais de mutação e cruzamento, e introduz um novo operador denominado remoção de atributos. Esse operador retira de um indivíduo todas as ocorrências de um determinado atributo, produzindo um novo indivíduo com um atributo selecionado a menos que o indivíduo original.

Sobre o método de geração da população inicial, não fica claro no artigo como isto foi realizado. Como método de seleção dos indivíduos a formarem as próximas gerações, foi utilizado o da roleta.

Para validar o SET-Gen, foram utilizadas 10 bases de dados públicas, as quais foram disponibilizadas em [29]. Os resultados mostraram que o SET-Gen diminuiu consideravelmente a complexidade das árvores induzidas, se comparada com o C4.5, tanto em relação ao tamanho quanto ao número de atributos selecionados. Quanto à acurácia, os resultados obtidos pelo SET-Gen foram levemente superiores. Entretanto, essa vantagem não foi estatisticamente significativa de acordo com o teste t [30].

3.2 Classificação de processos de negócio

O interesse inicial das organizações estava focado apenas na gestão e automação de processos. Para suprir essas necessidades, muitas soluções foram desenvolvidas, tais como CRM, SCM, ERP, BPM e WfMS. Porém, à medida que essas tecnologias foram se consolidando, novas necessidades e dificuldades foram surgindo.

As empresas não estão mais interessadas em apenas acompanhar e controlar o andamento de seus processos, mas sim em poder medi-los, associar valores à execução de cada instância de processo, atividade e recurso (tanto humano quanto computacional) e analisar seu desempenho. Além disso, elas buscam soluções que sejam capazes de prever eventuais anomalias de execução, que monitorem e disparem avisos de alerta ou notificações para as pessoas responsáveis e identifiquem eventuais áreas de melhoria no processo. De modo geral, o interesse das organizações passou a ser focado na análise, monitoração e previsão de seus processos de negócio [5–7]. A seguir, são apresentados dois trabalhos seminais na área de BI.

3.2.1 *Business Process Intelligence* (BPI)

Grigori *et al.* em [6] desenvolveram uma plataforma de software denominada BPI, a qual dá suporte à definição, à execução e ao acompanhamento de processos de negócio. BPI é composta por um conjunto de ferramentas integradas que dão suporte aos usuários, de negócio e de TI, na gestão da qualidade das execuções dos processos de negócio. Para isso, BPI provê diversas características como análise, predição, monitoramento, controle e otimização.

Uma das ferramentas disponibilizadas em BPI é o PME (*Process Mining Engine*), o qual permite um tipo mais "inteligente"de análise ao executar algoritmos de mineração nos dados históricos de execuções de processos de negócio. Tais dados são registrados por sistemas de gestão de *workflow*. O BPI é considerado pioneiro na utilização da técnica de classificação para análise de processos de negócio. Os modelos de classificação são representados através de árvores de decisão, as quais são construídas pelo algoritmo C4.5 [28]. A justificativa para a utilização de árvores de decisão é a facilidade de interpretação.

Como método de pré-processamento, é proposta em [6] a criação de uma tabela de análise de processos, a qual inclui uma linha para cada instância de processo e cujas colunas correspondem aos atributos da instância de processo. Uma coluna adicional é necessária para representar o atributo classe (presença ou não de um comportamento específico). Nesta etapa, os autores destacam a questão de quais atributos devem compor esta tabela, enfatizando o problema da presença de atividades que podem ser executadas inúmeras vezes. A indicação dos autores é que, sobre estas atividades, sejam armazenadas informações sobre sua primeira e última execução, além do número de vezes que foram executadas. A justificativa para isso é que estas informações estão mais correlacionadas com o atributo classe, de acordo com experimentos realizados. Maiores detalhes sobre quais informações devem compor esta tabela de análise de processos podem ser encontrados em [6].

É possível perceber que o número de atributos que compõem a tabela de análise de processos pode se tornar muito grande à medida que a complexidade do processo aumenta. Neste sentido, é destacada a importância da seleção de atributos. Porém, a técnica utilizada para tratar este problema é muito trivial, pois os atributos preditivos são avaliados independentemente quanto a suas relevâncias para o atributo classe. Estes atributos são ordenados de acordo com uma medida de avaliação (não especificada qual é utilizada) e os x primeiros da lista são selecionados, sendo x um valor pré-definido.

3.2.2 *Intelligence Business Operation Management* (iBOM)

O iBOM [7] é uma extensão do BPI e objetiva a definição de métricas, a análise inteligente, a predição e a otimização de operações de negócio. Ele acrescenta a idéia do APM (*Abstract*

Process Monitor), o qual é utilizado para a definição e gestão de processos abstratos. Esses processos abstratos representam visões, onde o usuário define quais atividades de um processo ele deseja analisar. Este sistema, diferentemente da abordagem do BPI, não interage apenas com sistemas de *workflow*, mas também possibilita a definição e monitoração de atividades, as quais podem estar dispersas em diversos sistemas heterogêneos.

Outro componente disponibilizado pelo iBOM é o FAPE (*Factor Analysis and Prediction Engine*). Sua função é aplicar técnicas de mineração nos dados históricos dos processos com o propósito de encontrar padrões no comportamento das métricas. Estes padrões são apresentados aos usuários na forma de árvores de classificação, as quais também são utilizadas para previsão dos valores das métricas de instâncias de processos em execução, proporcionando, de forma pró-ativa, a redução de um possível prejuízo no caso do valor previsto para esta métrica ser indesejável.

Em [7] é destacada a idéia de que, ao contrário da análise descritiva, o modelo de classificação utilizado para previsão não necessariamente deve ter como prioridade o critério de interpretabilidade, sendo mais importantes critérios como acurácia e desempenho. É mencionada a intenção de realização de experimentos com outros modelos (ex: SVM e redes bayesianas), sendo que, caso forneçam resultados satisfatórios, podem inclusive ser introduzidos no FAPE e utilizados para previsão.

O pré-processamento é o mesmo apresentado pelo BPI [6]. No entanto, é interessante destacar que foi observada uma maior preocupação com a técnica de seleção de atributos utilizada. Neste caso, foi utilizado o CFS (*Correlation-Based Feature Selection*) [31], o qual é um método mais sofisticado para seleção de atributos e utiliza geração heurística e avaliação de dependência (baseada em correlações).

4 Ferramenta MGAFS

Neste capítulo é apresentado o protótipo da ferramenta desenvolvida neste trabalho. É feita a descrição do protótipo bem como a validação do mesmo, de modo a tornar confiáveis os resultados obtidos pela ferramenta. No próximo capítulo, será descrita a utilização do protótipo para seleção de atributos em classificação de processos de negócio.

4.1 Descrição de MGAFS

4.1.1 Introdução

Neste trabalho foi desenvolvido, em linguagem Java, um protótipo denominado MGAFS (*Genetic Algorithm for Feature Selection*), que é a implementação de um algoritmo genético como método *wrapper* de seleção de atributos no contexto de algoritmos de classificação.

O princípio de funcionamento de MGAFS, apresentado na Figura 7, segue os passos:

1. Geração dos indivíduos que compõem a primeira geração da população. Cada indivíduo representa uma possível solução do problema, que neste caso é um subconjunto de atributos.

2. Cada indivíduo passa pelo processo de avaliação, onde são calculados seus respectivos valores de aptidão. Caracterizando o método de seleção de atributos como *wrapper*, a base de dados representada por um indivíduo é utilizada para gerar um classificador, o qual produz como saída as métricas para o cálculo do valor de aptidão (*fitness*). Essas bases de dados são compostas somente pelos atributos selecionados pelo indivíduo em questão, além do atributo classe.

3. O critério de parada é verificado. Caso seja satisfeito, o algoritmo mostra na saída o melhor indivíduo da população, o qual representa a solução do problema ($3b$ em Figura 7). Caso contrário, a população passa pelo processo de evolução, onde são realizadas operações de seleção, cruzamento, substituição e mutação ($3a$ em Figura 7). Após, o processo se repete a partir do passo 2. Cabe destacar que podem ser utilizados dois critérios de parada: número de evoluções da população ou se há algum indivíduo que possua um valor de *fitness* satisfatório.

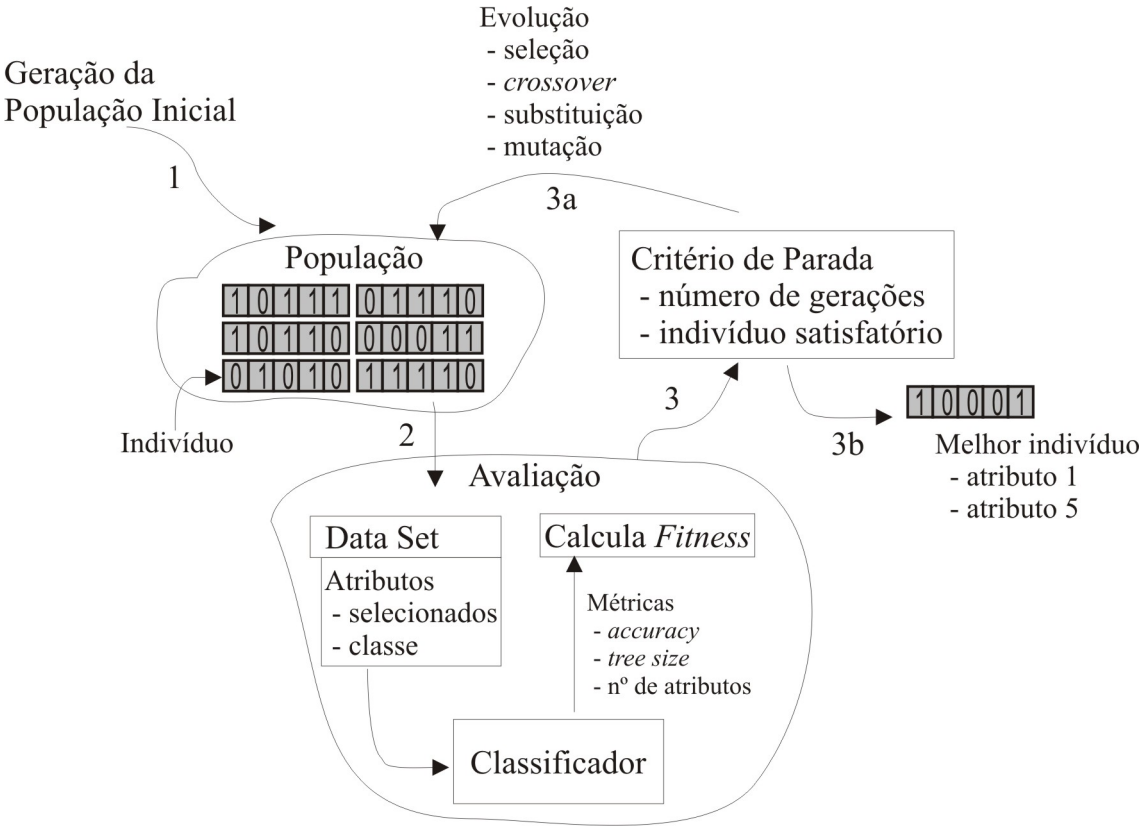

Figura 7: Algoritmo genético como método de seleção de atributos.

4.1.2 Parâmetros de configuração

O usuário pode configurar as seguintes opções do algoritmo genético:

- ***n_generation*:** Número de evoluções da população;
- ***n_individuals*:** Número de indivíduos que cada população possui;
- ***prob_crossover*:** A probabilidade de ocorrência de *crossover*;
- ***prob_mutation*:** A probabilidade de ocorrência de mutação.

4.1.3 Métodos implementados em MGAFS

MGAFS implementa estratégias simples para as diferentes técnicas relacionadas aos algoritmos genéticos. A seguir, são apresentadas as técnicas implementadas.

4.1.3.1 Codificação dos indivíduos

Cada cromossomo é representado por um *string* binário de comprimento igual ao número de atributos, o qual é obtido da base dec dados. Cada posição do *string* (gene do indivíduo) corresponde a um atributo da base. Quando o valor da posição i do *string* for 1, significa que o atributo i é selecionado. Caso contrário, significa que o atributo não é selecionado. Assim, cada indivíduo corresponde a um subconjunto de atributos selecionados para representar a base de dados que será utilizada para a construção do modelo de classificação. A Figura 8 ilustra um exemplo de cromossomo neste formato.

Figura 8: Cromossomo visto como um *string* binário.

4.1.3.2 Geração da população inicial

Para a geração da população inicial é utilizada uma técnica aleatória que elimina a probabilidade de os indivíduos possuírem um número de 1s sempre em torno da metade do número total de atributos. Para isso, é sorteado o número de 1s para cada indivíduo e, após, são sorteadas as posições no cromossomo que receberão 1. Este método é o mesmo proposto em [9].

4.1.3.3 Avaliação

A função de *fitness* utilizada para a avaliação de cada indivíduo x pode ser composta pelas seguintes métricas:

- acurácia do classificador (accuracy());
- complexidade do classificador (complexity()).
- número de atributos selecionados (numberFS()).

O peso para cada uma das métricas é atribuído pelo usuário. A Equação 4.1 ilustra um exemplo de função de *fitness*, onde a acurácia (accuracy()) recebe peso 100, a complexidade (complexity()) recebe peso -10 e o número de atributos selecionados (numberFS()) recebe peso igual a -5.

$$fitness(x) = 100 * accuracy(x) - 10 * complexity() - 5 * numberFS() \tag{4.1}$$

4.1.3.4 Método de seleção

Foram implementados os métodos *Ranking* e Roleta, os quais são apresentados a seguir.

a) Roleta

Este é o método de seleção mais comum, onde cada indivíduo possui um intervalo de números de uma roleta. Esse intervalo de números é proporcional ao valor de *fitness* do indivíduo. A roleta é girada n vezes, onde n é o número de indivíduos da população. A cada giro da roleta, um número z é marcado. O indivíduo selecionado é aquele que possuir o número z dentro de seu intervalo de números. No final do processo, são selecionados n indivíduos reprodutores.

Tabela 1: Representação dos valores de *fitness* para cada indivíduo de uma população.

Indivíduo	1	2	3	4	5	6	7	8	9	10
Valor de *Fitness*	7	6	1	10	4	8	3	3	6	9

Supondo a Tabela 1, a qual representa uma população de 10 indivíduos com seus respectivos valores de *fitness*, a roleta com os intervalos de números destinados a cada indivíduo pode ser visualizada na Figura 9.

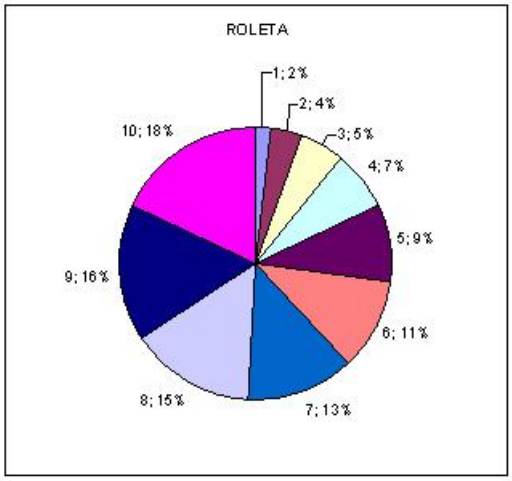

Figura 9: Roleta representando a proporção do intervalo de números reservado para cada um dos 10 (dez) indivíduos da população.

b) *Ranking*

O método *Ranking* possui como objetivo prevenir a convergência prematura para "ótimos locais"no espaço de soluções do problema. Nesse caso, os indivíduos da população são ordenados de acordo com o seu valor de aptidão e a probabilidade de escolha é atribuída conforme a posição do indivíduo no *ranking*. Neste caso, o problema de grandes diferenças entre as chances de seleção dos indivíduos é automaticamente eliminado, uma vez que a relação das probabilidades de escolha entre dois indivíduos de posições i e $i + 1$ no *ranking* é independente das diferenças absolutas de seus valores de aptidão, dependendo apenas das posições no *Ranking*.

O *Ranking* evita dar probabilidade de seleção maior para um subgrupo pequeno de indivíduos altamente aptos, reduzindo, assim, a pressão da seleção quando a variação da aptidão é muito elevada.

Se, por um lado, não levar em conta a informação de aptidão absoluta é vantajoso para evitar uma convergência prematura, por outro é desvantajoso, visto que, em alguns casos, pode ser importante conhecer a diferença entre os valores de aptidão de dois indivíduos consecutivos no *Ranking*, principalmente nas gerações finais da população, onde espera-se que estejam as melhores soluções para o problema.

4.1.3.5 Reprodução

Primeiramente, deve ser determinada uma forma de escolha dos pares de indivíduos (pai e mãe) que formarão seus descendentes (filhos). Após, deve ser estabelecido um métodos de cruzamento (*crossover*), o qual é utilizado pelos progenitores na reprodução de seus filhos.

Para a seleção dos pares foi implementado o método aleatório, no qual os pares de indivíduos são escolhidos ao acaso dentre aqueles selecionados na etapa de seleção.

Para *crossover*, foram implementados 1-*Point Crossover*, 2-*Point Crossover* e *Uniform Crossover*.

a) *1-Point Crossover*

Este tipo de cruzamento é composto por um ponto de *crossover* em uma posição p, onde o $filho_1$ recebe os genes do pai até o ponto p, e os genes da mãe do ponto p em diante. O $filho_2$, ao contrário, recebe os genes da mãe até o ponto p e os genes do pai do ponto p em diante. Esse processo pode ser visualizado na Figura 10.

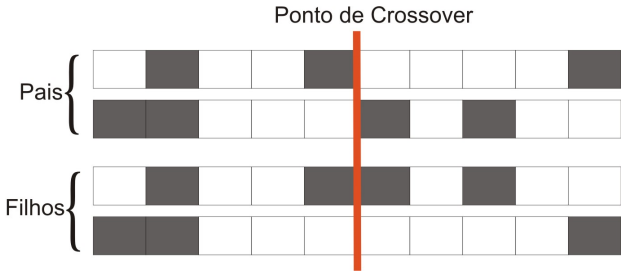

Figura 10: Cromossomos reproduzidos, filhos, após cruzamento dos pais através de 1 ponto.

b) *2-Point Crossover*

São utilizados 2 pontos de *crossover*, p_1 e p_2. Analogamente ao *crossover* de 1 ponto, os filhos são frutos do intercâmbio dos genes dos pais. O $filho_1$ recebe os genes do pai até o ponto p_1; do ponto p_1 até o ponto p_2 ele recebe os genes da mãe e do ponto p_2 em diante recebe os genes do pai novamente. O surgimento do $filho_2$ segue a mesma idéia e pode ser visualizado na Figura 11.

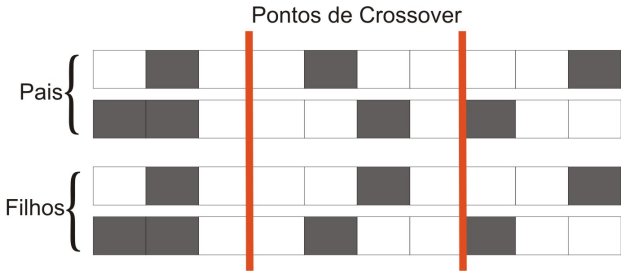

Figura 11: Cromossomos reproduzidos, filhos, após cruzamento dos pais através de 2 pontos.

c) *Uniform Crossover*

Inicialmente, os filhos $filho_1$ e $filho_2$ são cópias idênticas de seus pais. Após, para cada gene i ($1 \leq i \leq n$, onde n é o número total de genes do cromossomo) é testado, sob uma probabilidade igual a 0.5, quanto a troca. Caso haja troca, os genes de posição i de cada filho são trocados. No final, cada filho representa combinações de genes dos pais.

4.1.3.6 Substituição

O método de substituição implementado no MGAFS foi o de substituição imediata, no qual os descendentes substituem seus pais, havendo uma substituição completa da população.

Também foi implementada a teoria do elitismo, onde os x melhores indivíduos são copiados diretamente para a próxima geração da população. O valor de x é fornecido pelo usuário.

4.1.3.7 Mutação

Após todo o processo de *crossover* e substituição, antes de a nova população ser concretizada, os indivíduos ainda podem passar por mutações. A técnica de mutação implementada é apresentada a seguir:

```
Sortear um número x de pares de genes.
Para cada x
   Sortear dois genes diferentes (índices do cromossomo)
   Trocar seus valores
```

4.1.3.8 Critério de Parada

O critério de parada pode ser um dos seguintes:

- número de evoluções da população;
- indivíduo com um valor de *fitness* satisfatório.

O valor a ser utilizado no critério de parada é configurado pelo usuário.

4.2 Validação de MGAFS

4.2.1 Introdução

Esta seção tem como objetivo validar MGAFS como método de seleção de atributos em problemas de classificação. Para isto, foram realizados experimentos com conjuntos de dados bem conhecidos na literatura, possibilitando uma comparação com outros trabalhos relacionados.

Primeiramente, são apresentados os conjuntos de dados utilizados para experimentos com MGAFS. Os resultados obtidos, bem como uma discussão dos mesmos, são apresentados na seqüência. Por fim, são realizadas comparações com os resultados fornecidos em dois trabalhos relacionados [3, 4].

4.2.2 Conjuntos de dados públicos

Os conjuntos de dados selecionados para os experimentos são apresentados na Tabela 2. Eles foram extraídos da U*CI Machine Learning Repository* [19] e *UCI KDD Archive* [20].

Tabela 2: Sumário dos conjuntos de dados utilizados.

Conjunto de Dados	**Instâncias**	**Atributos**	**Classes**	**Acurácia** (%)	**Tamanho da Árvore**
Anneal	898	38	6	98, 44	47
Bands	540	39	2	70, 19	7
Credit-g	1000	20	2	70, 50	140
Credit-a	690	15	2	86, 09	42
Labor	57	16	2	73, 68	5
Colic	368	22	2	85, 33	6
Autos	205	26	7	81, 95	69
Arrhythmia	452	279	16	64, 38	99

4.2.3 Execução de MGAFS

O MGAFS foi executado com os seguintes parâmetros:

- ***n_generation*** = 10;
- ***n_individuals*** = 20;
- ***prob_crossover*** = 0,9;
- ***prob_mutation*** = 0,05.

Tabela 3: Resultados obtidos com a execução de MGAFS

Data Set	*Number of features*		*Tree Size*		*Accuracy*	
	MGAFS	Without FS	MGAFS	Without FS	MGAFS	Without FS
anneal	$20,20 \pm 3,27$	38	$37 \pm 7,97$	47	$98,20 \pm 1,17$	$98,44$
bands	$14,60 \pm 6,66$	39	$5,40 \pm 2,19$	7	$71,07 \pm 1,10$	$70,19$
credit-g	$8,62 \pm 4,65$	20	$2 \pm 2,23$	140	$70,34 \pm 0,76$	$70,50$
credit-a	$3,40 \pm 0,55$	15	3 ± 0	42	$85,51 \pm 0$	$86,09$
labor	$4,60 \pm 2,51$	16	$5,60 \pm 2,51$	5	$89,12 \pm 3,14$	$73,68$
colic	$6 \pm 1,87$	22	$7,60 \pm 2,07$	6	$86,41 \pm 0,64$	$85,33$
autos	$13,80 \pm 2,77$	26	$50,40 \pm 4,04$	69	$81,17 \pm 2,69$	$81,95$
arrhythmia	$31,80 \pm 11,97$	279	$69,40 \pm 12,76$	99	$65,62 \pm 3,99$	$64,38$

Os experimentos foram realizados em um computador Pentium 4 1.8Ghz e 384MB RAM DDR 333Mhz. A forma de representação escolhiada para o modelo de classificação foi árvores de decisão, e o algoritmo de indução o J48, o qual é disponibilizado no *software* Weka (www.cs.waikato.ac.nz/ml/weka/) e corresponde à implementação do algoritmo C4.5 proposto por Quinlan em [28]. Os parâmetros foram ajustados com os valores sugeridos pelo Weka.

Como o procedimento de geração dos indivíduos de um algoritmo genético trabalha de forma aleatória, para cada conjunto de dados foram realizadas 5 (cinco) execuções de MGAFS e, após, calculou-se a média dos resultados obtidos. Cabe salientar que esse número de execuções foi escolhido em razão de as referências [3] e [4] também o utilizarem, possibilitando uma comparação mais justa dos resultados. Para cada uma das 5 (cinco) execuções foram capturados, para cada conjunto de dados, o número de atributos selecionados, a número de nodos e a acurácia da árvore de decisão gerada pelo subconjunto de dados composto apenas pelos atributos selecionados. Cabe destacar que a acurácia é calculada por 10-*fold cross-validation* [22].

Os experimentos foram realizados com MGAFS configurado com diferentes métodos de seleção e *crossover*. Os melhores resultados foram obtidos com a seleção através da Roleta e com 2-*Point Crossover*.

Com o objetivo de desenvolver um estudo da contribuição dos diferentes critérios utilizados como função de *fitness*, empiricamente foram atribuídos diferentes pesos a estes critérios e os melhores resultados foram encontrados com a função representada na Equação 4.2 que, por esse motivo, será a utilizada nos experimentos deste trabalho. A análise que comprova a superioridade desta função é descrita por Basgalupp *et al.* em [32], artigo publicado em *International Workshop on Feature Selection for Data Mining: Interfacing Machine Learning and Statistics*.

$$fitness = accuracy(x) - 0.2(nfs(x) + c(x)) \tag{4.2}$$

onde $accuracy(x)$ representa a acurácia do classificador gerado pelo cromossomo x, $nfs(x)$ indica o número de atributos selecionados no cromossomo x e $c(x)$ representa a complexidade do classificador, a qual foi representada pelo número de nodos da árvore gerada por x.

A Tabela 3 apresenta os resultados médios (antes de $\pm$), com os respectivos desvios-padrão (após $\pm$), obtidos nas execuções de MGAFS para cada conjunto de dados.

Como se pode observar, houve acréscimo na acurácia para os conjuntos de dados Bands, Labor, Colic e Arrhythmia (desconsiderando-se os desvios-padrões). Embora não tenha aumentado a acurácia para todos os conjuntos de dados, exceto para Colic, onde o tamanho da árvore é menor sem a utilização de MGAFS, para todos os outros conjuntos de dados o MGAFS diminuiu tanto o número de atributos selecionados quanto o tamanho das árvores geradas. Isso se deve à inclusão desses critérios na função de *fitness*, a qual é utilizada com o intuito de abrir mão de uma maior acurácia em prol de maior interpretabilidade do modelo de classificação. Basgalupp *et al*. destacam essa idéia em [32], onde também apresentam resultados para uma função de *fitness* composta apenas pela acurácia e, nesse caso, MGAFS aumenta a acurácia para todos os conjuntos de dados, porém com valores menos satisfatórios para os outros dois critérios (número de atributos e tamanho da árvore de decisão). Cabe destacar que a redução da acurácia para as bases de dados Anneal, Credit-g, Credit-a e Autos não foi significativa de acordo com o teste *t* [30].

4.2.4 Comparação com trabalhos relacionados

Alguns dos conjuntos de dados utilizados nestes experimentos também já foram usados em outros trabalhos. Isso permite que seja realizada uma comparação superficial entre os resultados obtidos nesses trabalhos e os aqui apresentados. A seguir, são apresentados dois trabalhos que também utilizaram algoritmos genéticos para seleção de atributos. As especificações dos experimentos realizados nestes trabalhos, bem como seus resultados, também são descritos.

4.2.4.1 Yang & Honavar

Foi utilizado um algoritmo genético para a seleção de atributos dos conjuntos de dados Anneal, Credit-g e Colic. Foram realizadas 5 (cinco) execuções do algoritmo genético para cada base de dados. Como classificador foram utilizadas redes neurais artificiais, sendo fornecidas por elas as acurácias calculadas com 10-*fold cross-validation*. A configuração do algoritmo genético nas execuções foi a seguinte:

- ***n_generation*** = 20;
- ***n_individuals*** = 50;
- ***prob_crossover*** = 0,6;
- ***prob_mutation*** = 0,01.

Uma comparação entre os resultados obtidos por Yang & Honavar em [3] e os obtidos por MGAFS são ilustrados na Tabela 4.

Tabela 4: Comparação dos resultados obtidos em [3] com os obtidos por MGAFS

	Atributos (nº)		Acurácia (%)	
	Yang & Honavar	MGAFS	Yang & Honavar	MGAFS
Anneal	$21 \pm 3,1(15)$	$20,2 \pm 3,27(15)$	$99,5 \pm 0,9$	$98,2 \pm 1,17$
Credit-a	$8 \pm 2,1(15)$	$3,4 \pm 0,55(15)$	$91,5 \pm 2,8$	$85,51 \pm 0$
Colic	$11 \pm 2,3(22)$	$6 \pm 1,87(22)$	$92,6 \pm 3,4$	$86,41 \pm 0,64$

É possível observar uma vantagem da utilização de MGAFS no que diz respeito ao número de atributos selecionados. Essa melhoria pode ser nitidamente visualizada para as bases de dados Credit-a e Colic, sendo que para Anneal há uma equivalência de resultados ao levar-se em consideração os desvios-padrões.

Entretanto, a acurácia do classificador usado em [3] é superior à apresentada pelo MGAFS para Credit-a e Colic, mesmo nos limites dos desvios-padrão. Já em Anneal pode-se considerar uma equivalência devido s desvios-padrão. Contudo, uma comparação das acurácias obtidas em cada um dos trabalhos não é justa, pois foram utilizados classificadores diferentes e é sabido que as redes neurais são mais precisas que as árvores de decisão.

Em relação ao tempo de execução não foi possível realizar uma comparação, pois em [3] não foram fornecidos estes tempos. Também não foi possível verificar se os atributos selecionados foram os mesmos, pois não foram fornecidas essas informações.

4.2.4.2 Cantú-Paz

Foram utilizados dois algoritmos genéticos (*sGA* e *FilterGA*) para a seleção de atributos dos conjuntos de dados Anneal e Arrhythmia. O que diferencia um do outro é o método de geração da população inicial, onde é proposta a utilização da saída de um método de seleção de atributos *filter* para tal.

Foram realizadas 5 (cinco) execuções do algoritmo genético para cada base de dados. Como classificador foi utilizada a técnica Naive Bayes. Para as execuções, os parâmetros de configurações de *sGA* e *FilterGA* utilizados foram os seguintes:

- ***n_individuals*** = $3\sqrt{t}$, onde t é o número de atributos;
- ***prob_crossover*** = 1;
- ***prob_mutation*** = $\frac{1}{t}$.

Os resultados obtidos por *FilterGA* e *sGA*, comparados com o MGAFS, são ilustrados na Tabela 5 e na Tabela 6, respectivamente.

A primeira observação que se faz ao visualizar as Tabelas 5 e 6 é que o método *FilterGA* é superior ao *sGA* em todos os requisitos. Entretanto, a superioridade do *FilterGA* sobre o MGAFS não é observada, visto que a acurácia de MGAFS é superior para os dois conjuntos de

Tabela 5: Comparação entre os resultados obtidos pelo método *FilterGA* em [4] e os obtidos por MGAFS

	Anneal		Arrhythmia	
	FilterGA	MGAFS	FilterGA	MGAFS
Atributos (nº)	$12,90 \pm 2,04(38)$	$20,20 \pm 3,27(38)$	$86,20 \pm 6,42(279)$	$31,80 \pm 11,97(279)$
Acurácia (%)	$93,07 \pm 2,89$	$98,20 \pm 1,17$	$64,16 \pm 2,13$	$65,62 \pm 3,99$

Tabela 6: Comparação entre os resultados obtidos pelo método sGA em [4] e os obtidos por MGAFS

	Anneal		Arrhythmia	
	sGA	MGAFS	sGA	MGAFS
Atributos (nº)	$22,10 \pm 3,81(38)$	$20,20 \pm 3,27(38)$	$138,90 \pm 4,99(279)$	$31,80 \pm 11,97(279)$
Acurácia (%)	$92,47 \pm 1,69$	$98,20 \pm 1,17$	$59,78 \pm 3,51$	$65,62 \pm 3,99$

dados. Além disso, o número de atributos selecionados por MGAFS é menor para o conjunto de dados Arrhythmia. Entretanto, como já havia sido mencionado anteriormente, uma análise quanto à previsão não é totalmente confiável quando os classificadores utilizados não são os mesmos.

Quando comparados os métodos *sGA* e MGAFS, observam-se vantagens do segundo em relação ao primeiro tanto no número de atributos selecionados quanto na acurácia do classificador. Já no número de atributos selecionados para Anneal, há uma equivalência em detrimento dos desvios-padrão.

4.2.5 Conclusões

Os algoritmos genéticos se mostraram eficazes quando aplicados à seleção de atributos em problemas de classificação. Isto foi constatado não somente nos resultados fornecidos por MGAFS, mas também nos obtidos em trabalhos relacionados apresentados.

Quando realizada uma comparação dos resultados obtidos por MGAFS com os obtidos nos trabalhos relacionados, observou-se no mínimo uma equivalência nos resultados. Isso indica que MGAFS fornece resultados confiáveis e que pode ser utilizada para o propósito deste trabalho, classificação de processos de negócio.

5 Mineração de processos de negócio

Neste capítulo são apresentados experimentos da utilização de MGAFS sobre dados reais de execuções de processos de negócio. Além de verificar a eficácia do protótipo na otimização dos critérios propostos na função de *fitness*, também se tem como objetivo detectar os problemas encontrados nos modelos de classificação gerados, principalmente no que diz respeito à interpretação dos mesmos.

5.1 Fonte de dados

5.1.1 Modelo de processo

O processo utilizado nesta pesquisa descreve o fluxo de um sistema de solicitações de desenvolvimento de *software*. Esse processo pertence a uma empresa real e contempla desde a descrição do módulo a ser implementado até o seu desenvolvimento e validação. A modelagem do processo segue as características presentes na ferramenta Oracle *Workflow*.

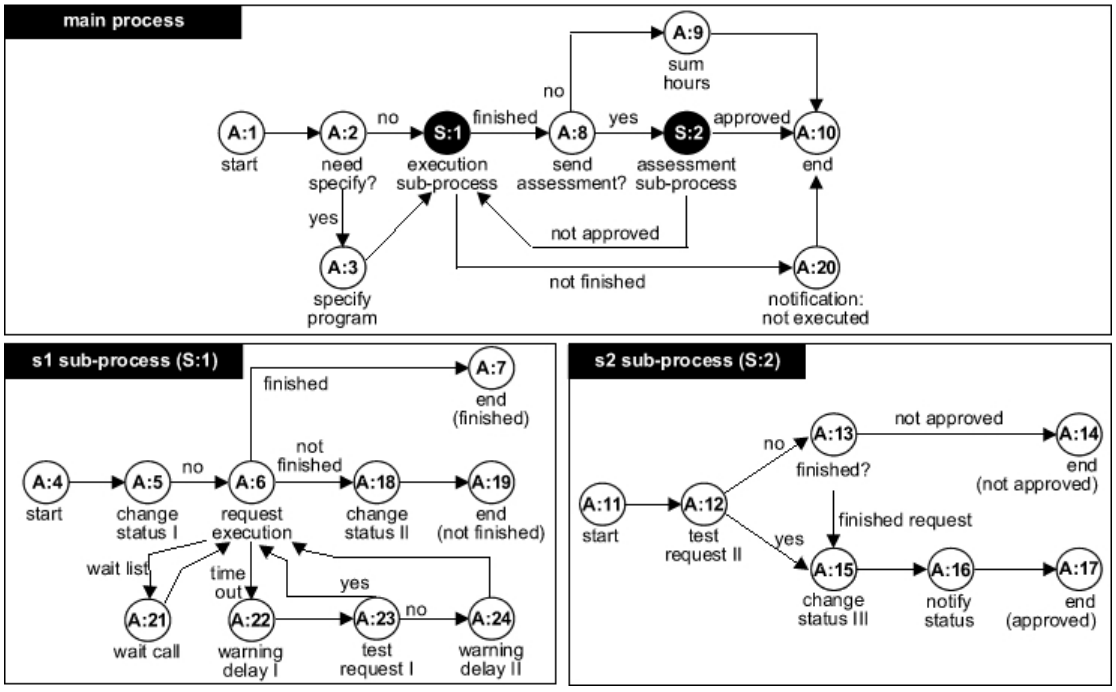

Figura 12: Fluxo do Sistema de Solicitações.

A Figura 12 ilustra o modelo do processo, o qual é composto por um processo principal (*main process*) e dois sub-processos, um responsável pela execução da solicitação, S:1, e outro pela avaliação da execução solicitada, S:2. A seguir, tanto o processo principal quanto os sub-processos são descritos em função do fluxo das atividades, sem entrar em maiores detalhes com relação à lógica utilizada.

Primeiramente, é verificada a necessidade de gerar a especificação do componente solicitado. Caso seja necessário, antes de executar o pedido, o solicitante deve especificar o programa a ser desenvolvido. Se o responsável pela implementação do componente constatar que não é possível atender os requisitos solicitados, o sistema envia uma mensagem notificando essa indisponibilidade da solicitação. Do contrário, após a implementação do componente, pode existir a necessidade de avaliá-lo. Se não for preciso avaliar o componente, o pedido é registrado como encerrado e as horas de trabalho são totalizadas. Sendo necessário avaliar o componente, o processo pode seguir diferentes caminhos. Se a avaliação for de reprovação, a solicitação é enviada novamente para o responsável pela execução (implementação), senão, o fluxo do processo termina.

No sub-processo S1, o pedido tem seu *status* alterado para "em andamento", A5, e a solicitação é encaminhada para o responsável, A6. Ele pode implementar o componente solicitado (A7), verificar a indisponibilidade de atender os requisitos solicitados (o pedido é registrado como encerrado e as horas de trabalho são totalizadas, A18), ou ainda, inserir o pedido numa lista de espera, A21. Caso o responsável exceda o tempo determinado para a execução da solicitação, *time out*, o solicitado e o solicitante recebem um aviso de atraso do pedido (A22, A23 e A24).

No sub-processo S2, primeiramente é verificado se o solicitado é a mesma pessoa que realizou a solicitação (A12). Caso positivo, o pedido é encerrado, as horas de trabalho são totalizadas e o solicitado recebe uma avaliação de aprovação do pedido (A15, A16 e A17). Caso contrário, estas tarefas somente serão realizadas se, mediante avaliação (A13), o solicitante validar o pedido. Do contrário, a solicitação é encaminhada novamente para a atividade de execução do pedido (A14).

Cabe salientar que não existem outros detalhes fora os comentados nesta seção sobre o processo ilustrado na Figura 12, o que prejudicou um pouco as atividades subseqüentes do processo de KDD.

Em meio a tantas alternativas de fluxo dentro do processo, deseja-se compreender quais são os fatores que contribuem para o tempo de execução do processo. Assim, formula-se o seguinte objetivo de negócio: Classificar as instâncias desse processo de acordo com o seu tempo de execução, buscando as características que determinam se o processo é terminado a tempo ou atrasa o final de sua execução.

5.1.2 Modelo de dados

Os dados utilizados foram disponibilizados em banco de dados Oracle. Essa base, ilustradas nas Figuras 13 e 14, é referente ao modelo de *Data Warehouse* proposto por Fábio Casati em [33]. Este modelo analítico foi alimentado a partir do modelo de dados dos registros de execuções do *Oracle Workflow* como parte do estudo de caso desenvolvido em [2].

A base de dados representa um modelo dimensional do tipo constelação, o qual é composto por tabelas fato e dimensão associadas [34]. Nas tabelas do tipo dimensão estão armazenados os dados principais referentes aos processos, atividades, recursos, etc. As tabelas do tipo fato armazenam as instâncias de processos, atividades, dados e serviços, as quais se relacionam com diferentes tabelas dimensão.

Como o objetivo é classificar processos de acordo com o tempo de execução dos mesmos, é necessário verificar se há dados disponíveis para atender esse objetivo. Assim, à medida que as tabelas forem sendo analisadas, serão feitos comentários sobre essa questão.

A definição dos processos disponíveis é armazenada na tabela dimensão PROC_DEFS_D, ilustrada na Figura 13, a qual contém informações como o nome do processo, versão e o grupo ao qual ele pertence.

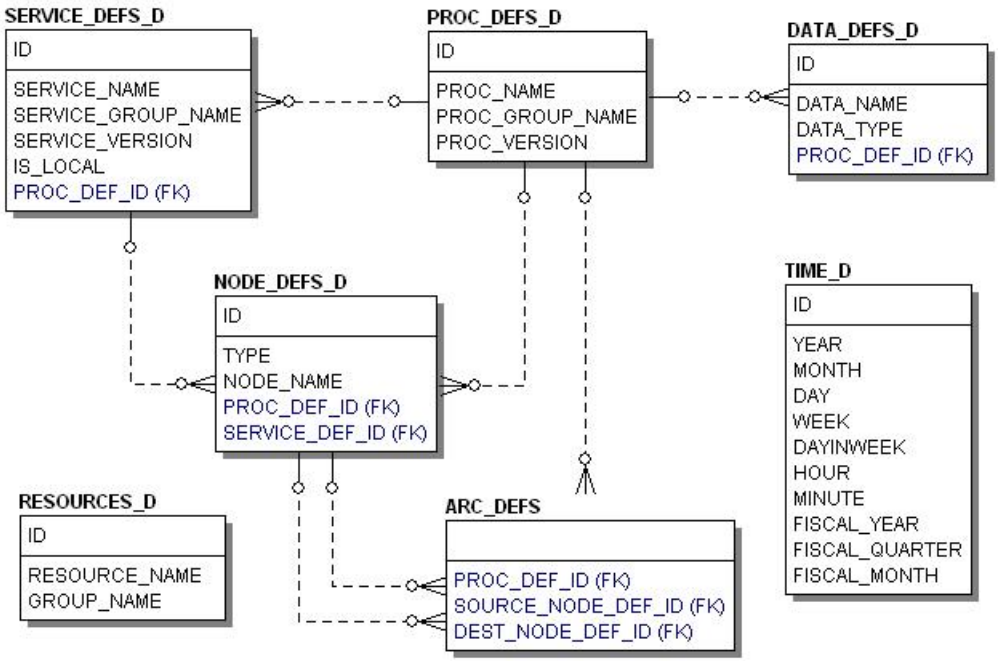

Figura 13: Dimensões definidas pelo Modelo Analítico. Fonte: [2].

As instâncias dos processos são armazenadas na tabela fato PROC_INST_F (Figura 14). Pode-se observar a presença de informações importantes como o tempo de início e de término do processo, tempo de duração desses processos em dias, horas e minutos e a pessoa responsável pela execução da instância do processo. Ou seja, é possível obter informações temporais para processos, o que é um bom indício de que os dados atendem ao objeto de negócio.

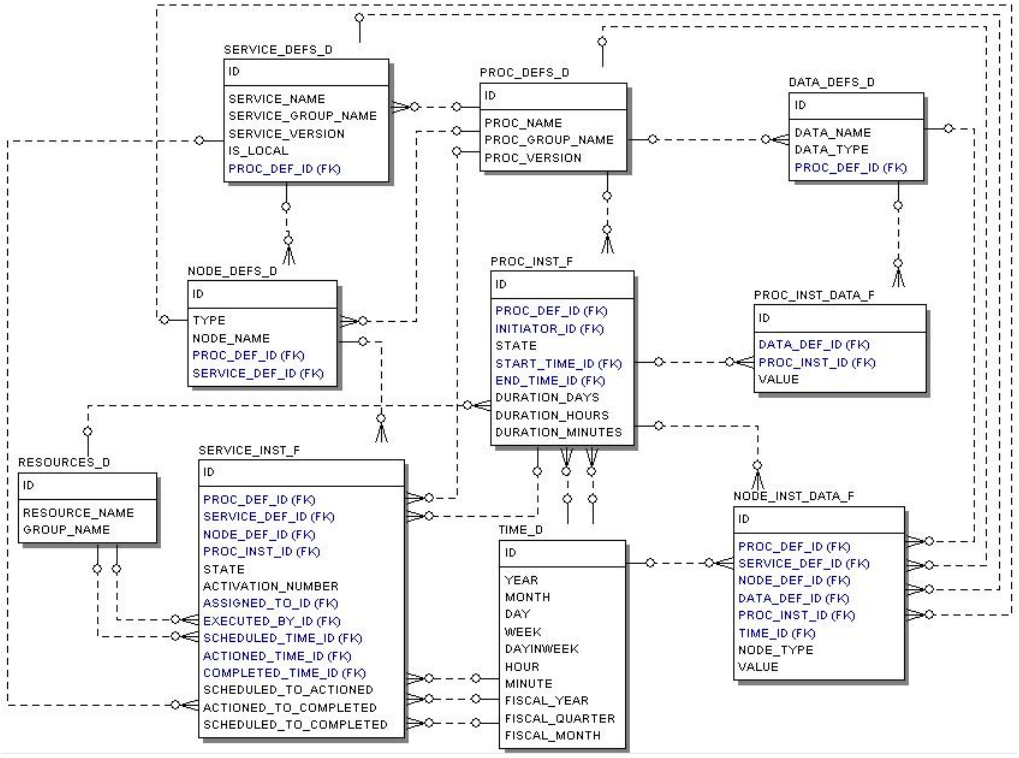

Figura 14: Fatos definidos pelo modelo analítico e seus relacionamentos. Fonte: [2].

Os processos podem possuir dados de controle associados a eles. Esses dados de controle são definidos na tabela DATA_DEFS_D e suas instâncias são representadas pela tabela PROC_INST_DATA_F (Figura 14). Tais dados de controle também podem influenciar no tempo de execução dos processos.

As definições das atividades são representadas pela tabela NODE_DEFS_D, ilustrada na Figura 13. Cada atividade pertence a uma definição de processo (chave estrangeira PROC_DEF_ID) e possivelmente possui um serviço associado (chave estrangeira SERVICE_DEF_ID). As atividades que não possuem um serviço associado são aquelas atividades de controle de fluxo, por exemplo, a atividade *start*. As atividades que possuem serviço associado são aquelas que interessam para este trabalho, pois demandam tempo e recursos para sua execução.

Na tabela SERVICE_INST_F estão as instâncias dos serviços. Nessa tabela constam informações tais como: a qual atividade pertence esse serviço, em qual definição e qual instância de processo o serviço é utilizado. Através dessa tabela também é possível saber se o serviço foi executado pelo mesmo recurso pelo qual ele foi programado, além de informar o tempo de execução esperado e o tempo que realmente foi utilizado para execução. Os serviços também possuem informações temporais, o que é bastante relevante para o objetivo de negócio proposto.

As atividades também podem conter dados. Esses dados, como, por exemplo, a descrição da tarefa realizada, são definidos na tabela DATA_DEFS_D e suas instâncias são armazenadas na tabela NODE_INST_DATA_F. As tabelas são ilustradas na Figura 14. Os dados referentes às atividades também podem ser úteis para a determinação do tempo de execução dos processos.

Os tempos são representados na tabela dimensão TIME_D. Eles podem ser expressos em várias unidades como, por exemplo, dia, semana, mês, ano, horas e minutos. Nota-se que essa tabela tem relacionamento com as tabelas SERVICE_INST_F e PROC_INST_F. Assim, pode-se concluir que as informações temporais sobre instâncias de processos e instâncias de serviços podem ainda ser mais detalhadas através da tabela TIME_D.

A partir do estudo realizado sobre o modelo para análise de processos de negócio, é possível concluir que o mesmo nos disponibiliza dados interessantes para atingir o objetivo de negócio. É verdade que isso não depende somente do modelo, mas sim dos dados com os quais o modelo é carregado.

Terminada essa etapa, a seguir será apresentada a preparação dos dados para a aplicação das técnicas de mineração.

5.2 Pré-processamento dos dados

Como etapa de pré-processamento dos dados, foi criado um *script* PL/SQL para gerar dinamicamente uma tabela de análise de processos, onde as linhas representam as instâncias de processos e as colunas correspondem aos atributos referentes tanto a informações de atividades como dados do processo como um todo. Os atributos que compõem essa tabela são os mesmos sugeridos em [6] para classificação de processos de negócio.

Adicionalmente, foi criado um atributo para representar a classe, onde cada instância (linha) do processo foi classificada como NO_PRAZO (terminou até o prazo máximo estabelecido) e FORA_DO_PRAZO (ultrapassou o prazo estabelecido para término). Das 1036 instâncias de processo, 648 foram classificadas como NO_PRAZO e 388 não terminaram até o tempo esperado (FORA_DO_PRAZO). Além do atributo classe, 141 atributos compuseram a tabela (24 numéricos e 117 categóricos). Por questões de confidencialidade, os nomes dos recursos, pessoas que executam atividades, foram alterados para nomes fictícios.

5.3 Experimentos

Inicialmente, foi gerada uma árvore de decisão sem a aplicação de MGAFS. Essa árvore foi induzida a partir do algoritmo J48 do Weka. Com os parâmetros configurados de acordo com o padrão do *software*, a árvore final foi composta por apenas um nodo folha representando a classe, com uma acurácia de 62,55%. Na realidade, essa acurácia representa a percentagem de instâncias de processo, no conjunto de treino, que são classificadas como NO_PRAZO, ou seja, a árvore nos diz que é melhor "arriscar predizer"que o processo terminará no prazo esperado, sem mesmo realizar qualquer tipo de análise. Em todos os experimentos, a acurácia é calculada

por 10-*fold cross-validation*.

Dando seguimento à experimentação, foi desabilitada a opção de poda da árvore nas configurações do algoritmo J48 e o mesmo foi re-executado. Como resultado, encontrou-se uma árvore de decisão com 63,8% de acurácia mas com 1028 nodos. Esse enorme número de nodos torna a árvore inviável de ser interpretada, justificando a opção do algoritmo J48 em "arriscar predizer"que o processo terminará no prazo, visto que não vale a pena utilizar uma árvore tão grande para obter um ganho de aproximadamente 1% na acurácia. Cabe salientar que esse grande número de nodos se deve à presença de atributos categóricos com muitos valores possíveis. Quando estes atributos são escolhidos pelo algoritmo, é criada uma aresta para cada um de seus valores, tornando a árvore de decisão muito complexa.

Após, com os valores padrões para os parâmetros do algoritmo J48, foram realizados vários experimentos com MGAFS, onde os parâmetros foram bastante variados. Empiricamente, os melhores resultados foram obtidos com a seguinte configuração:

- número de gerações: 50;
- tamanho da população: 50;
- probabilidade de mutação: 0.05;
- probabilidade de *crossover*: 0.9;
- método de *crossover*: 2-*Point Crossover*;
- método de seleção: Roleta;
- *fitness* = *accuracy*(x) - 0.2(*nfs*(x) + *c*(x)).

Foram realizadas 30 execuções de MGFAS com a configuração supracitada, e o resultado (média e desvio padrão) obtido é apresentado na Tabela 7.

Tabela 7: Execução de MGFAS para o conjunto de dados de execução de processos de negócio.

	Sem Seleção de Atributos	MGAFS
Número de Atributos	141	$4,55 \pm 2,20$
Tamanho da Árvore	1028	$21,03 \pm 3,03$
Acurácia	$63,80$	$72,79 \pm 1,10$

É possível observar a melhoria dos critérios analisados. Nota-se que, embora não seja uma taxa muito alta de acurácia, houve um aumento considerável. O número médio de atributos selecionados também foi baixo, porém o desvio padrão é alto em relação à magnitude do valor médio. Já para o tamanho médio da árvore houve uma melhoria bastante significativa, visto que viabilizou a interpretação das regras.

Em uma das 30 execuções de MGAFS, foram selecionados 9 dos 141 atributos preditivos, originando uma árvore de decisão composta de 24 nodos e uma acurácia de 74,13%. As regras derivadas são apresentadas na Figura 15 .

```
P_DURATION_DAYS <= 0: NO_PRAZO
P_DURATION_DAYS > 0:
|| AT_SOLICITANTE = JOAO: NO_PRAZO
|| AT_SOLICITANTE = MANOEL: ATRASADO
|| || DURATION48332P <= 13: ATRASADO
|| || DURATION48332P   > 13: NO_PRAZO
|| AT_SOLICITANTE = JOSE: NO_PRAZO
|| AT_SOLICITANTE = JOAQUIM
|| || P_DURATION_DAYS <= 33: NO_PRAZO
|| || P_DURATION_DAYS   > 33: ATRASADO
|| AT_SOLICITANTE = CARLOS: NO_PRAZO
|| AT_SOLICITANTE = DIEGO: NO_PRAZO
|| AT_SOLICITANTE = MARCIO: NO_PRAZO
|| AT_SOLICITANTE = LEANDRO: ATRASADO
|| AT_SOLICITANTE = DIOGO: NO_PRAZO
|| AT_SOLICITANTE = TOBIAS: NO_PRAZO
|| AT_SOLICITANTE = RICARDO: NO_PRAZO
|| AT_SOLICITANTE = JORGE: NO_PRAZO
|| AT_SOLICITANTE = RAFAEL: ATRASADO
|| AT_SOLICITANTE = THIAGO: NO_PRAZO
|| AT_SOLICITANTE = ROBERTO: ATRASADO
|| AT_SOLICITANTE = GUSTAVO: ATRASADO
|| AT_SOLICITANTE = CRISTIAN: NO_PRAZO
```

Figura 15: Regras que representam a árvore de decisão produzida após uma das execuções de MGFAS.

Observa-se que há algumas regras bastante interessantes. Uma delas é: *os processos com pelo menos um dia de duração, solicitados por MANOEL, terminam no prazo quando a atividade A3, a qual realiza a especificação do* software *em desenvolvimento, é executada em mais de 13 dias (DURATION48332P), caso contrário os processos acabam atrasando*. Uma possível explicação para isso é que os processos mal especificados (considera-se "mal especificados", aqui, aqueles processos em que a especificação é realizada rapidamente) apresentam problemas posteriores que comprometem a entrega do processo no prazo correto.

Pode parecer um pouco óbvio utilizar informações temporais como atributos preditivos. Em outras palavras, pode parecer óbvio que, se uma atividade X levar 2 meses para ser executada, o processo terminará fora do prazo porque não há processos que duram mais de 1 mês. Com o intuito de encontrar regras que relacionam recursos, bem como outras informações do ambiente, para com o atributo classe, foram removidos todos os atributos que correspondem a tempos de duração. Também foram excluídos atributos de descrição, preenchidos com linguagem natural, cujos valores são quase todos distintos. Isso resultou em uma base de dados com 98 atributos preditivos, a qual gerou uma árvore de decisão bastante interessante, composta por 27 nodos e uma acurácia de 69,79%. Essa árvore, em formato de regras, é apresentada na Figura 16.

A partir desse resultado pode-se encontrar outras regras interessantes, como por exemplo: *se o processo é solicitado (AT_SOLICITANTE) por JOAO e o início desse processo é realizado até o dia 25 do mês, então ele terminará no prazo. Caso contrário, o processo atrasa*. Há casos

```
AT_SOLICITANTE = JOAO
|| P_START_DAY <= 25: NO_PRAZO
|| P_START_DAY   > 25: ATRASADO
AT_SOLICITANTE = MANOEL: ATRASADO
AT_SOLICITANTE = JOSE
|| P_START_DAY <= 25: NO_PRAZO
|| P_START_DAY   > 25: ATRASADO
AT_SOLICITANTE = JOAQUIM
|| AT_URL_PACKAGE = prod: NO_PRAZO
|| AT_URL_PACKAGE = http://localhost/IP_Soil: ATRASADO
|| AT_URL_PACKAGE = m: ATRASDO
AT_SOLICITANTE = CARLOS: NO_PRAZO
AT_SOLICITANTE = DIEGO
|| P_END_HOUR <= 8: ATRASADO
|| P_END_HOUR   > 8: NO_PRAZO
AT_SOLICITANTE = MARCIO: NO_PRAZO
AT_SOLICITANTE = LEANDRO: ATRASADO
AT_SOLICITANTE = DIOGO: NO_PRAZO
AT_SOLICITANTE = TOBIAS: NO_PRAZO
AT_SOLICITANTE = RICARDO: NO_PRAZO
AT_SOLICITANTE = JORGE: NO_PRAZO
AT_SOLICITANTE = RAFAEL: ATRASADO
AT_SOLICITANTE = THIAGO: NO_PRAZO
AT_SOLICITANTE = ROBERTO: ATRASADO
AT_SOLICITANTE = GUSTAVO: ATRASADO
AT_SOLICITANTE = CRISTIAN: NO_PRAZO
```

Figura 16: Regras que representam a árvore de decisão produzida após uma das execuções de MGFAS.

em que basta apenas saber quem solicitou o processo para inferir o atraso ou não do processo. É o caso do RAFAEL, onde a árvore nos diz o processo solicitado por ele atrasa. Esse tipo de informação é interessante para que o especialista de negócio faça uma análise das razões pelas quais os processos solicitados por RAFAEL atrasam.

Certamente se houvesse a ajuda de um especialista de negócio seria possível realizar uma análise mais detalhada do significado das regras encontradas. Porém, como já mencionado, a falta desse recurso prejudicou esse tipo de análise.

A seguir, são apresentados dois problemas detectados, os quais são específicos para a classificação de processos de negócio e, como não podem ser resolvidos na etapa de seleção de atributos, são apenas apresentados e sugeridos como trabalhos futuros.

5.3.1 Problemas detectados

Para ilustrar os problemas encontrados, foi gerada uma base de dados fictícia para representar possíveis execuções do processo representado pela Figura 17. A Tabela 8 ilustra uma amostra destes dados, onde é armazenado, para cada atividade, apenas o recurso humano que a executou. Há também a presença de um atributo extra, indicando qual caminho foi escolhido no ponto de decisão do processo. Por fim, cada instância (ID) foi classificada (CLASSE) como NO_PRAZO e FORA_DO_PRAZO.

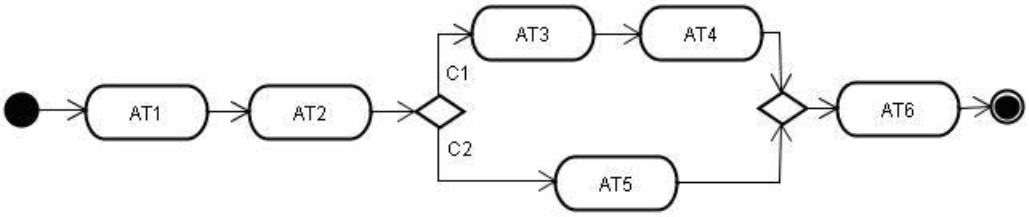

Figura 17: Exemplo de um processo de negócio simples.

Tabela 8: Base de dados que representa possíveis execuções do processo representado na Figura 17

ID	AT1	AT2	EXTRA	AT3	AT4	AT5	AT6	CLASSE
1	KARIN	DUNCAN	C1	KARIN	DUNCAN	?	MÁRCIO	NO_PRAZO
2	KARIN	MÁRCIO	C1	MÁRCIO	KARIN	?	DUNCAN	NO_PRAZO
3	KARIN	KARIN	C2	?	?	MÁRCIO	DUNCAN	FD_PRAZO
4	KARIN	DUNCAN	C2	?	?	MÁRCIO	KARIN	FD_PRAZO
5	KARIN	DUNCAN	C2	?	?	KARIN	DUNCAN	FD_PRAZO
6	MÁRCIO	KARIN	C1	MÁRCIO	DUNCAN	?	KARIN	NO_PRAZO
7	MÁRCIO	DUNCAN	C1	MÁRCIO	KARIN	?	DUNCAN	NO_PRAZO
8	MÁRCIO	MÁRCIO	C1	DUNCAN	KARIN	?	KARIN	FD_PRAZO
9	MÁRCIO	DUNCAN	C1	DUNCAN	MÁRCIO	?	MÁRCIO	FD_PRAZO
10	MÁRCIO	KARIN	C1	KARIN	DUNCAN	?	MÁRCIO	NO_PRAZO

5.3.1.1 Caminhos alternativos

Como podem haver caminhos alternativos nos fluxos dos processos, algumas atividades ora podem ser realizadas e ora não. São os casos das atividades AT3, AT4 e AT5, denominadas atividades alternativas.

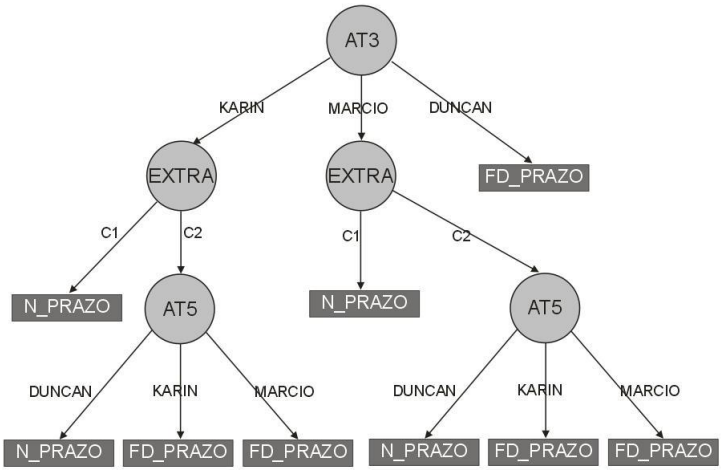

Figura 18: Árvore de decisão que gerada pelo algoritmo J48 para os dados da Tabela 8.

Quando uma instância de processo segue pelo lado C1, a atividade AT5 não é executada e, portanto, não há valor associado a qualquer dado referente a ela. Quando o caminho C2 é seguido, as atividades AT3 e AT4 não são executadas e, analogamente, também não há dado sobre essas atividades. Assim, pode ocorrer que a árvore de decisão gerada como modelo de

classificação utilize um atributo referente a uma atividade alternativa para representar um nodo da árvore. É o caso da árvore de decisão da Figura 18, a qual foi gerada pelo algoritmo J48 do Weka, onde AT3 aparece no nodo raíz da árvore. Sendo assim, este modelo de classificação se torna inútil tanto para análises preditivas como descritivas de instâncias de processos que seguirem por C2, visto que não há como chegar a um nodo folha caso passe por qualquer nodo que teste algum atributo referente à atividade AT3. Trabalhos como [6] e [7] não apresentam alternativa para tratar este problema e nem mesmo o identificam.

5.3.1.2 Ordem das atividades

No intuito de utilizar o modelo de classificação para previsão de comportamentos de instâncias de processos que ainda estão em execução, há a idéia de o classificador representar a ordem de ocorrência das atividades. Isto porque, em determinados pontos no fluxo do processo, só estão disponíveis dados referentes às execuções das atividades que antecedem esse ponto. Assim, é desejável que a ordem de execução das atividades seja coerente com a ordem de busca utilizada ao se percorrer uma árvore de decisão até chegar a um nodo folha, o qual identifica o atributo classe.

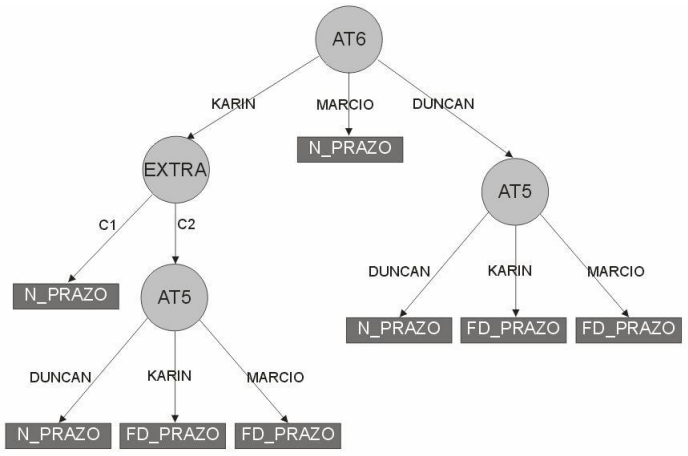

Figura 19: Suposta árvore de decisão para representar os dados da Tabela 8.

Supondo a árvore representada pela Figura 19, como o nodo raiz testa o recurso que executou a atividade AT6, não há como tentar prever o comportamento de qualquer instância de processo que ainda não tenha chegado a esta atividade. Provavelmente, quando esta atividade entrar em execução, não haverá mais tempo de tomar qualquer atitude caso algum problema seja detectado. Este problema não inviabiliza análises descritivas, porém torna as árvores de decisão inúteis para previsão de comportamento de processos ainda em execução, desejo este manifestado em trabalhos como [6] e [7].

5.3.1.3 Possíveis soluções

Para resolver o problema dos caminhos alternativos, uma possível solução seria incluir o atributo EXTRA (o que já pôde ser observado no exemplo anterior), representando o caminho seguido no fluxo do processo. Porém, como já visto, não basta apenas inserir este atributo. Seria também necessário garantir que este atributo extra esteja presente antes (percorrendo a árvore a partir da raiz) das atividades alternativas, o que não se garante com a utilização do algoritmo C4.5, por exemplo.

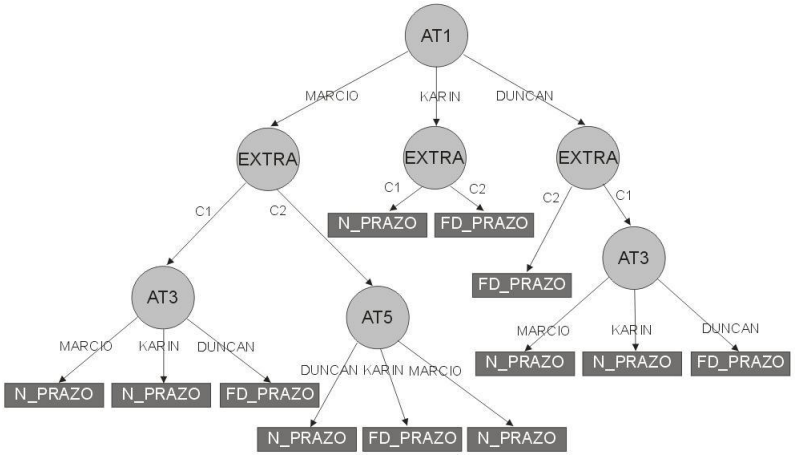

Figura 20: Árvore de decisão considerada ideal para representar os dados da Tabela 8.

Já o problema da ordem das atividades é algo mais complexo de ser tratado e deve receber muita atenção. De alguma maneira, deve-se dar prioridade aos atributos referentes à atividades que são executadas primeiro no fluxo, estando assim esta árvore mais coerente com a ordem de execução.

Tais problemas devem ser resolvidos no algoritmo de indução do modelo de classificação, através da variação de algum algoritmo existente (C4.5, por exemplo) ou então com a construção de um novo algoritmo específico para classificação de processos de negócio, ou de outros domínios que possuem características dessa natureza. Com isso, espera-se que seja gerada uma árvore de decisão como a representada na Figura 20, a qual representa fielmente os dados da Tabela 8 e não apresenta os problemas detectados.

6 Considerações Finais

6.1 Contextualização

Trabalhos da área de Inteligência de Negócio (*Business Intelligence*) têm enfatizado a necessidade dos gestores em ter um melhor controle de suas operações e de como as mesmas estão relacionadas aos objetivos do negócio. Nesse contexto, é destacado o papel das técnicas de mineração de dados para a análise, previsão e otimização de processos de negócio. Para isso, a técnica de classificação é bastante utilizada, onde o objetivo é entender as causas de determinados comportamentos e gerar modelos para análises descritivas e preditivas do comportamento e do desempenho destes processos.

Porém, a quantidade de atributos que caracterizam um processo pode ser enorme, pois, além dos atributos diretamente relacionados a uma instância de processo, também devem ser considerados os atributos pertencentes às atividades contidas nesse processo. Dependendo da disposição das atividades no fluxo de processo, algumas delas podem ser executadas inúmeras vezes (ciclos), aumentando ainda mais o número de atributos a serem considerados, e outras podem nem mesmo serem executadas (caminhos alternativos), gerando uma grande quantidade de dados faltantes. Estes fatores implicam baixa qualidade dos modelos resultantes, alto custo computacional e, principalmente, dificuldade de interpretação dos modelos.

Neste contexto, insere-se o objetivo deste trabalho, que é propor um método *wrapper* de seleção de atributos baseado em algoritmos genéticos multiobjetivos. Dentre as vantagens desta aplicação, tem-se a possibilidade de otimização de diferentes critérios e não de um único, que geralmente é a acurácia do classificador. Além disso, trabalhos como [6] e [7] utilizam métodos *filter* de seleção de atributos, mas tais métodos são bastante simples e não apresentam resultados satisfatórios quando o problema é mais complexo [1].

6.2 Resultados obtidos

Com os resultados obtidos neste trabalho, pode-se constatar a aplicabilidade dos algoritmos genéticos como método multiobjetivo de seleção de atributos para problemas de classificação de processo de negócio. Os critérios utilizados na função de *fitness* foram melhorados quando comparados aos resultados sem a utilização da seleção de atributos.

Embora não tenha sido detalhado o pré-processamento para a classificação de processos de

negócio, o tempo consumido nessa etapa foi bastante grande, cerca de 60% do total utilizado durante o processo de descoberta de conhecimento. Isso comprava o que foi constatado por Fayyad *et al*. em [35], onde é prevista a maior utilização de tempo na etapa de pré-processamento.

Mesmo sem a presença de um especialista de negócio, o que enriqueceria bastante as regras de negócio encontradas, pode-se concluir que essas regras podem ajudar a empresa, a qual forneceu os dados, a melhorar o andamento de seus processos, principalmente através de relocação de recursos.

6.3 Projeto XEN

A ferramenta MGAFS também pode ser utilizada em outros domínios de aplicação. Como exemplo, tem-se o projeto HP-XEN, onde há o interesse da aplicação do processo de descoberta de conhecimento sobre *benchmarks* de execuções de máquinas virtuais XEN.

A idéia principal é utilizar os resultados da descoberta de conhecimento para encontrar melhores configurações do ambiente de execução, de forma que certas métricas tenham melhores desempenhos. Assim, foi criado um problema de classificação, onde as métricas são rotuladas como BOM_DESEMPENHO e MAU_DESEMPENHO e, então, cada execução gera um *benchmark* que pode ser classificado de acordo com a métrica escolhida.

No projeto, já foi desenvolvido um modelo analítico como etapa prévia à mineração de dados. Quando se constrói a tabela de classificação dos *benchmarks*, há um grande número de atributos preditivos, o que sugere a utilização de MGAFS.

Resultados iniciais mostraram que vale a pena investir nesta linha de pesquisa, pois, mesmo com poucos dados carregados no modelo analítico, já foi possível obter regras interessantes no contexto de execução de máquinas virtuais XEN. E, à medida que o modelo for recebendo mais dados, MGFAS poderá se tornar ainda mais útil.

6.4 Trabalhos futuros

A primeira necessidade seria a aquisição de uma nova base de dados de registros de execução de *workflow*, bem como a ajuda de pelo menos um especialista de negócio, com o intuito de realizar uma análise mais profunda ao nível de negócio.

Há o interesse na evolução de MGAFS, incluindo novas implementações dos diferentes métodos (ex: geração da população inicial, *crossover* e mutação) do algoritmo genético, inserindo novos critérios (ex: custo de cada atributo) na função de *fitness*, e criando uma interface gráfica mais amigável, ajudando o usuário.

Entretanto, o principal interesse como trabalho futuro, o que inclusive já resultou em um

projeto para o doutorado, é a criação de um algoritmo de indução de árvore de decisão específica para classificação de processos de negócio, resolvendo os problemas detectados no presente trabalho: caminhos alternativos e ordem das atividades.

Referências

[1] DASH, M.; LIU, H. Feature selection for classification. **Intelligent Data Analysis**, Elsevier, V. 1, n. 1-4, p. 131-156, July 1997.

[2] GARCIA, R. S.; RUIZ, D. D. Pré-Processamento de Dados para Descoberta de Conhecimento em Processos de Workflow Modelados sobre Plataforma Oracle. In: ESCOLA REGIONAL DE BANCO DE DADOS (ERBD 2005), 1., abr. 2005, Porto Alegre. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, abr. 2005, p. 25-30.

[3] YANG, J.; HONAVAR, V. Feature Subset Selection Using a Genetic Algorithm. In: GENETIC PROGRAMMING: ANNUAL CONFERENCE, 2., 13-16 July 1997, Califórnia. **Proceedings...** Califórnia: Stanford University, July 1997, p. 13-16.

[4] CANTÚ-PAZ, E. Feature Subset Selection, Class Separability, and Genetic Algorithms. In: THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE (GECCO), 6., 26-30 June, Seattle. **Proceedings...** Seattle: Red Lion Hotel, June 2004, p. 959-970.

[5] SUN, Z.; BEBIS, G.; YUAN, X.; LOUIS, S. Genetic Feature Subset Selection for Gender Classification: A Comparison Study. In: IEEE WORKSHOP ON APPLICATIONS OF COMPUTER VISION, 6., 3-4 Dec. 2002, Orlando. **Proceedings...** Orlando: IEEE Computer Society, Dec. 2002, p. 165-170.

[6] CHERKAUER, K.; SHAVLIK, J. Growing Simpler Decision Trees to Facilitate Knowledge Discovery. In: INTERNACIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2., 2-4 Aug. 1996, Portland. **Proceedings...** California: AAAI Press, Aug. 1996, p. 315-318.

[7] GOLFARELLI, M.; RIZZI, S.; CELLA, L. Beyond data warehousing: what's next in business intelligence? In: ACM INTERNATIONAL WORKSHOP ON DATA WAREHOUSING AND OLAP, 7., Nov. 2004, Washington. **Proceedings...** New York: Data Warehousing and OLAP, Nov. 2004, p. 1-6.

[8] GRIGORI, D.; CASATI, F.; CASTELLANOS, M.; DAYAL, U.; SAYAL, M.; SHAN, M. C. Business Process Intelligence. **Computers in Industry**, Elsevier, v. 53, n. 3, p. 321-343, Apr. 2004.

[9] CASTELLANOS, M.; CASATI, F.; MING-CHIEN S.; Dayal, U. iBOM: A Platform for Intelligent Business Operation Management. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE 2005), 21., 5-8 Apr. 2005, Tokyo. **Proceedings...** Tokyo: IEEE Computer Society, Apr. 2005, p. 1084-1095.

[10] TAN, P.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. Boston: Addison Wesley, 2005, 769 p. 72

[11] **The Workflow Management Coalition Specification - Terminology & Glossary: WFMC-TC-1011**, Winchester: Workflow Management Coalition, Feb. 1999, p. 65.

[12] BASGALUPP, M. **Uso de classificação para análise de processos de negócio**. jul. 2005, 79 f. Trabalho Individual I (Mestrado em Ciência da Computação)-Faculdade de Ciência da Computação, PUCRS, Porto Alegre, 2005.

[13] KOTSIANTIS, S.; PINTELAS, P. On the selection of classifier-specific feature selection algorithms. I**JSIT Lecture Note of International Conference on Intelligent Knowledge Systems**, Assos, v. 1, p. 153-160, Aug. 2004.

[14] LIU, H.; MOTODA, H. **Feature Selection for Knowledge Discovery and Data Mining**. Norwell: Kluwer Academic Publishers, 1998, 244 p.

[15] COELLO, C. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. **Knowledge and Information Systems**, Toronto, v. 1., n. 3, p. 129-156, Feb. 1999.

[16] GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston: Addison Wesley, 1989, 432 p.

[17] MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge: MIT Press, 1998, 224 p.

[18] YANG, J.; PAREKH, R.; HONAVAR, V. DistAl: An inter-pattern distance-based constructive learning algorithm. **Intelligent Data Analysis**, Ottawa, v. 3, n. 1, p. 55-73, Dec. 1999.

[19] NEWMAN, D.; HETTICH, S.; BLAKE, C.; MERZ, C. UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. Disponível em: <http://www.ics.uci.edu/ mlearn/MLRepository.html>. Acessado em: 20 mar. 2005.

[20] HETTICH, S.; BAY, S. The UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science. Disponível em: <http://kdd.ics.uci.edu>. Acessado em: 20 mar. 2005.

[21] RICHELD, M.; LANZI, P. Performing effective feature selection by investigating the deep structure of the data. In: INTERNACIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2., 2-4 Aug. 1996, Portland. **Proceedings...** California: AAAI Press, Aug. 1996, p. 379-383.

[22] HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco: Morgan Kaufmann, 2001. 550 p.

[23] DUDA, R.; HART, P.; STORK, D. **Pattern Classification, Second Edition**. California: Wiley-Interscience, 2000, 654 p.

[24] VAPNIK, V. **The nature of statistical learning theory**. New York: Springer-Verlag, 1995, 188p.

[25] VALENTIN, D.; ABDI, H.; EDELMAN, B.; O'TOOLE, A. Principal Component and Neural Network Analyses of Face Images: What Can Be Generalized in Gender Classification? **Mathematical Psychology**, Bloomington: Academic Press IDEAL, v. 41, p. 398-412, July 1997.

[26] ESHELMAN, L. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In: PROCEEDINGS OF THE WORKSHOP ON FOUNDATIONS OF GENETIC ALGORITHMS, 1., 15-18 July 1990, Bloomington. **Proceedings...** San Francisco: Morgan Kaufmann, July 1990, p. 265-283.

[27] PAPPA, G. **Seleção de atributos utilizando algoritmos genéticos multiobjetivos**. 2002. 85 f. Dissertação (Mestrado em Informática Aplicada) - Centro de Ciências Exatas e Tecnologia, PUC-PR, Curitiba, 2002.

[28] QUINLAN, J. **C4.5: programs for machine learning**. San Francisco: Morgan Kaufmann, 1993, 316p.

[29] BURL, M.; FAYYAD, U.; PERONA, P.; SMYTH, P.; BURL, M. Automating the Hunt for Volcanoes on Venus. In: PROCEEDINGS OF CONFERENCE ON COMPUTER VISITION & PATTERN RECOGNITION, 13., 21-23 Jun. 1994, Seattle. **Proceedings...** Seattle: IEEE Computer Society, Jun. 1994, p. 302-309.

[30] WONNACOTT, T.;WONNACOTT, R. **Introdução à Estatística**. Rio de Janeiro: Editora S.A: Livros Técnicos e Científicos, 1980, 589 p.

[31] HALL, M. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In: PROCEEDINGS OF INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 17., 29 Jun - 2 July, 2000, Palo Alto. **Proceedings...** San Francisco: Morgan Kaufmann, July 2000, p. 359-366.

[32] BASGALUPP, M.; BECKER, K.; RUIZ, D. A Study of multi-objective fitness functions for a feature selection genetic algorithm. In: PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON FEATURE SELECTION FOR DATA MINING: INTERFACING MACHINE LEARNING AND STATISTICS, 2., 22 Apr., 2006, Bethesda. **Proceedings...** Bethesda: Columbus, Apr. 2006, p. 123-130.

[33] CASATI, F. **Intelligent Process Data Warehouse for HPPM 5.0: HPL-2002-120**, HP Laboratories Palo Alto: Software Technology Laboratory, 26 Apr. 2002.

[34] KIMBALL, R.; REEVES, L.; THORNTHWAITE, W.; ROSS, M. **The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom**. New York: John Wiley, 1998, 800 p.

[35] FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: INTERNACIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2., 2-4 Aug. 1996, Portland. **Proceedings...** California: AAAI Press, Aug. 1996, p. 82-88.