SELEÇÃO AUTOMÁTICA DE EXEMPLOS DE TREINO PARA UM MÉTODO DE DEDUPLICAÇÃO DE REGISTROS BASEADO EM PROGRAMAÇÃO GENÉTICA

GABRIEL SILVA GONÇALVES

SELEÇÃO AUTOMÁTICA DE EXEMPLOS DE TREINO PARA UM MÉTODO DE DEDUPLICAÇÃO DE REGISTROS BASEADO EM PROGRAMAÇÃO GENÉTICA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: ALBERTO HENRIQUE FRADE LAENDER
CO-ORIENTADOR: MARCOS ANDRÉ GONÇALVES

Belo Horizonte
Abril de 2010

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Seleção automática de exemplos de treino para um método de deduplicação de registros baseado em programação genética

GABRIEL SILVA GONÇALVES

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ALBERTO HENRIQUE FRADE LAENDER - Orientador
Departamento de Ciência da Computação - UFMG

PROF. MARCOS ANDRÉ GONÇALVES - Co-orientador
Departamento de Ciência da Computação - UFMG

PROF. ANTÔNIO DE PÁDUA BRAGA
Departamento de Engenharia Eletrônica - UFMG

PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 30 de abril de 2010.

Aos meus pais e irmão, Carlos, Gislene e Mateus.

Agradecimentos

Gostaria de agradecer primeiramente aos meus pais, Carlos e Gislene, por todo o apoio dado durante todos estes anos. Não fosse por eles, sempre conscientes da importância dos estudos para a formação intelectual, moral e espiritual do ser humano, eu dificilmente estaria concluindo esta importante etapa da minha vida acadêmica.

Agradeço ao professor Alberto Henrique Frade Laender, meu orientador, pela oportunidade de trabalhar com uma pessoa tão competente e atenciosa, grande responsável pela realização de todas as etapas deste trabalho de dissertação. Ao professor Marcos André Gonçalves, meu co-orientador, que também contribuiu de forma bastante significativa para a realização deste trabalho. Ao Moisés, colega do Laboratório de Bancos de Dados (LBD) que acaba de terminar o seu doutorado, responsável pelo ensinamento de boa parte do conhecimento teórico necessário para a realização deste trabalho.

Ao Mateus, meu irmão, que possivelmente seguirá o mesmo caminho acadêmico que nosso pai e eu seguimos. A toda minha família, responsável por me tornar uma pessoa cada vez melhor e mais responsável. Ao Luciano Romero, um amigo que sempre esteve disposto a ajudar no meu mestrado e na minha vida acadêmica. Aos meus amigos Rodrigo, Diogo, Fábio, Izabella e Francina, que me acompanham por mais de (ou quase) uma década, sendo grandes responsáveis por tornar o caminho até aqui muito mais fácil.

Agradeço também aos excelentes professores que tive em Viçosa e aos inúmeros amigos que fiz durante a graduação, principalmente ao Marcelo, Leandro, Thiago, Fabrício e Douglas, pelos incontáveis momentos de bagunça e estudos. Aos meus amigos Felipe, Henrique, Luís Felipe e Leonardo, que me ajudaram durante esses dois anos de mestrado. Ao LiverNull, nosso time de futebol, responsável por inúmeras derrotas e quedas na tabela de classificação. Ao amigos do Synergia e do LBD, pela convivência extremamente harmoniosa e divertida.

A Deus, por tudo.

Resumo

O grande volume de informação disponível em meios digitais tem preocupado administradores de grandes repositórios de dados, tais como bibliotecas digitais e bancos de dados de grandes corporações. Atualmente, é possível estabelecer uma relação entre a qualidade dos dados presentes nos sistemas de uma organização e a capacidade da mesma prover serviços de qualidade a seus clientes, resultando em um grande investimento por parte de empresas e instituições governamentais no desenvolvimento de métodos eficientes para a identificação e remoção de réplicas. Por ser uma tarefa que exige muito tempo e poder de processamento, os métodos propostos devem obter bons resultados da forma mais eficiente possível.

Recentemente, técnicas de aprendizado de máquina vêm sendo utilizadas para lidar com o problema de deduplicação de registros. No entanto, elas exigem exemplos, normalmente gerados manualmente, para a realização da etapa de treino necessária para o aprendizado dos padrões de duplicação do repositório de dados. Isto dificulta a utilização dessas técnicas em diversos casos, devido ao custo exigido para a criação do conjunto de exemplos de treino.

Esta dissertação propõe uma abordagem que utiliza uma técnica determinística para sugerir automaticamente exemplos de treino para um método de deduplicação de registros baseado em programação genética (PG). Experimentos utilizando dados sintéticos mostram que é possível utilizar conjuntos de treino bastante reduzidos para se gerar mais rapidamente as funções de deduplicação, sem uma redução significativa na qualidade das soluções geradas, mesmo em repositórios de dados com elevados níveis de dificuldade para deduplicação. Além disso, foi realizado um projeto fatorial para mensurar o grau de dificuldade para se deduplicar repositórios de dados, identificando as características que podem afetar a utilização do método de seleção de exemplos de treino para a deduplicação de registros baseada em PG.

Palavras-chave: Identificação de Duplicatas, Inteligência Artificial, Programação Genética.

Abstract

The increasing volume of information available in digital media is becoming a challenge for administrators of large data repositories such as digital libraries and databases of large corporations. Nowadays, it is possible to say that the quality of the data used by an organization is proportional to its capacity of providing useful services to their users. Thus, companies and government institutions are investing a lot of money in developing efficient methods to identify and remove duplicates in large data repositories. Because record deduplication is a task that demands a lot of time and processing power, the proposed methods should be able to get good results as efficiently as possible.

Recently, machine learning techniques have been used to deal with the record deduplication problem. However, these techniques require examples – usually generated manually – to perform a training phase necessary to learn duplication patterns from existing data, what may restrict the use of such techniques due to the cost required to create the training set.

This MSc thesis proposes an approach that uses a deterministic technique to automatically suggest training examples for a record deduplication method based on genetic programming (GP). Experiments using synthetic data show that it is possible to use reduced training sets to faster generate deduplication functions without significantly reducing the quality of the solutions generated, even in data repositories with high levels of difficulty for deduplication. In addition, a factorial design was performed to measure the difficulty levels to deduplicate data repositories, identifying the characteristics that may affect the use of our approach to selecting training examples for the record deduplication method based on GP.

Keywords: Replica Identification, Artificial Intelligence, Genetic Programming.

Lista de Figuras

2.1 Exemplo de uma Função de Deduplicação mapeada como Árvore. 10
2.2 Exemplo de Execução da Operação de Cruzamento. 12
2.3 Exemplo de Execução da Operação de Mutação. 12
2.4 Esquema do Processo de Deduplicação de Registros utilizando a Abordagem baseada em Programação Genética. 19

3.1 Exemplo de Pares de Registros Positivos e Pares de Registros Negativos em um Repositório de Dados composto por Seis Registros. 24

4.1 Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros para os Experimentos que utilizam o Repositório MF ("Mais Fácil"). . . . 37
4.2 Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros para os Experimentos que utilizam o Repositório MD ("Mais Difícil"). . . 37
4.3 F1 Médio e Desvio Padrão do Melhor Indivíduo, nos Arquivos de Teste A, B e C, respectivamente, para os Experimentos que utilizam o Repositório MF ("Mais Fácil"). 38
4.4 F1 Médio e Desvio Padrão do Melhor Indivíduo, nos Arquivos de Teste A, B e C, respectivamente, para os Experimentos que utilizam o Repositório MD ("Mais Difícil"). 39
4.5 Representação Visual das Médias dos Percentuais de Variação Explicada pelos Fatores e Interações mais Relevantes do Projeto Fatorial $2^4 \times 30$, cujos valores são apresentados na Tabela 4.7. 50

5.1 Tela da Ferramenta em Desenvolvimento – Aba Referente à Configuração dos Parâmetros da PG. 53

Lista de Tabelas

3.1 Redução Percentual de Pares de Registros Positivos – Tempos de Treino, Médias e Desvios Padrões de F1. 29

3.2 Redução Percentual de Pares de Registros Negativos – Tempos de Treino, Médias e Desvios Padrões de F1. 30

3.3 Redução Percentual de Pares de Registros Positivos e Pares de Registros Negativos – Tempos de Treino, Médias e Desvios Padrões de F1. 31

3.4 Utilizando o Método de Fellegi & Sunter [1969] para a Geração do Conjunto de Treino – Tempos de Treino, Médias e Desvios Padrões de F1. 32

4.1 Características dos Repositórios Utilizados para a Realização dos Experimentos Iniciais: Repositório MF ("Mais Fácil") e Repositório MD ("Mais Difícil"). 34

4.2 Redução de Pares de Registros utilizando o Repositório MF ("Mais Fácil") – Tempos de Treino, Médias e Desvios Padrões de F1. 35

4.3 Redução de Pares de Registros utilizando o Repositório MD ("Mais Difícil") – Tempos de Treino, Médias e Desvios Padrões de F1. 36

4.4 Níveis dos Fatores utilizados no Projeto Fatorial. 42

4.5 Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros, em cada Experimento do Projeto Fatorial 2^4, com 100% e 25% dos Exemplos de Treino. 46

4.6 F1 Médio e Desvio Padrão do Melhor Indivíduo em cada Arquivo de Teste (A, B e C), para os Experimentos que utilizam 100% e 25% dos Exemplos de Treino. 49

4.7 Valores dos Percentuais de Variação Explicada pelos Fatores e Interações mais Relevantes do Projeto Fatorial $2^4 \times 30$, utilizando 100% e 25% dos Exemplos de Treino. 50

A.1 Avaliação do Impacto no Tempo de Treino – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **100% e 25% dos Exemplos de Treino**. 56
A.2 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste A**, utilizando **100% dos Exemplos de Treino**. 57
A.3 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste B**, utilizando **100% dos Exemplos de Treino**. 58
A.4 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste C**, utilizando **100% dos Exemplos de Treino**. 59
A.5 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste A**, utilizando **25% dos Exemplos de Treino**. 60
A.6 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste B**, utilizando **25% dos Exemplos de Treino**. 61
A.7 Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste C**, utilizando **25% dos Exemplos de Treino**. 62
A.8 Avaliação do Impacto na Qualidade das Soluções Geradas – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **100% dos Exemplos de Treino**. 63
A.9 Avaliação do Impacto na Qualidade das Soluções Geradas – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **25% dos Exemplos de Treino**. 63
A.10 Avaliação do Impacto na Qualidade das Soluções Geradas – Intervalos de Confiança (Limites Inferior e Superior) e Nível Máximo de Confiança para os Efeitos de cada Fator e Interação do Projeto Fatorial, em cada Arquivo de Teste (A, B e C), utilizando **100% dos Exemplos de Treino**. 64
A.11 Avaliação do Impacto na Qualidade das Soluções Geradas – Intervalos de Confiança (Limites Inferior e Superior) e Nível Máximo de Confiança para os Efeitos de cada Fator e Interação do Projeto Fatorial, em cada Arquivo de Teste (A, B e C), utilizando **25% dos Exemplos de Treino**. 64

Sumário

Agradecimentos ix

Resumo xi

Abstract xiii

Lista de Figuras xv

Lista de Tabelas xvii

1 Introdução 1

1.1 Motivação 2

1.2 Contribuição 3

1.3 Trabalhos Relacionados 3

1.3.1 Abordagens *Ad-hoc* 4

1.3.2 Abordagens baseadas em Treino 4

1.4 Organização da Dissertação 7

2 Deduplicação de Registros baseada em Programação Genética 9

2.1 Programação Genética 9

2.1.1 Operações Genéticas 11

2.1.2 Parâmetros de Controle 13

2.1.3 Algoritmo Evolucionário Geracional 15

2.2 Visão Geral da Abordagem para Deduplicação de Registros baseada em Programação Genética 17

2.2.1 Deduplicação de Registros utilizando a Abordagem baseada em PG 19

2.2.2 Análise de Complexidade do Processo de Geração das Funções de Deduplicação 21

3 Abordagem para Seleção Automática de Exemplos de Treino **23**
3.1 Definição de Pares de Registros 23
3.2 Seleção Automática de Exemplos de Treino 24
3.3 Resultados Experimentais 26
3.3.1 Conjunto de Dados Experimental 27
3.3.2 Descrição dos Experimentos 28

4 Avaliação da Abordagem Proposta **33**
4.1 Experimentos Iniciais 34
4.2 Projeto Fatorial 40
4.2.1 Impacto na Qualidade das Soluções Geradas 42
4.2.2 Impacto no Tempo de Treino 46
4.2.3 Conclusões Finais do Projeto Fatorial 48

5 Conclusões e Trabalhos Futuros **51**

Apêndice A Tabelas do Projeto Fatorial **55**

Referências Bibliográficas **65**

Capítulo 1

Introdução

O volume de dados coletados e armazenados pelas empresas vem aumentando de forma bastante significativa. Segundo uma pesquisa realizada pela Enterprise Strategy Group[1], uma renomada empresa de análise de mercado, 13 por cento das médias empresas pesquisadas no ano de 2004 armazenavam e utilizavam mais de 10 terabytes de dados. Em 2008, esse número subiu para 42 por cento, ou seja, mais que triplicou em um intervalo de tempo de apenas quatro anos [Geer, 2008].

Devido ao aumento do volume de informação disponível em meios digitais, administradores de grandes repositórios de dados, tais como bibliotecas digitais e bancos de dados de grandes corporações, vêm encontrando problemas para manter a qualidade dos dados disponíveis em seus repositórios. Uma vez que os repositórios de dados geralmente são construídos e complementados através da integração entre diferentes fontes de dados, é possível que ocorram diversas inconsistências, permitindo a geração indesejada de dados duplicados, resultando em repositórios com dados "sujos".

Atualmente, é possível estabelecer uma relação entre a qualidade dos dados presentes nos sistemas de uma organização e sua capacidade de prover serviços de qualidade a seus clientes. A decisão de manter repositórios com dados "sujos" vai além de questões técnicas, como desempenho e qualidade dos sistemas que utilizam esses dados. Além de esforços técnicos, também são necessárias mudanças culturais e no gerenciamento dos dados [Bell & Dravis, 2006].

A manutenção de repositórios com dados "sujos" pode acarretar diversos problemas. Por exemplo, o desempenho de um sistema gerenciador de banco de dados possivelmente será afetado, uma vez que dados adicionais sem utilidade demandam um maior processamento, exigindo mais tempo para responder consultas simples feitas pelos usuários. A qualidade das informações extraídas do repositório de dados também

[1] http://www.enterprisestrategygroup.com/

será prejudicada, pois a presença de réplicas e inconsistências pode gerar distorções em relatórios, levando à tomada de decisões incorretas. Além disso, é esperado que ocorra um aumento de custos operacionais, já que uma elevação (desnecessária) no volume de dados acarreta maiores investimentos em mídias de armazenamento e poder de processamento computacional, visando a manutenção dos tempos de resposta aos usuários em níveis aceitáveis.

O problema de detectar e remover entradas duplicadas em repositórios de dados é conhecido como *deduplicação de registros* [Koudas et al., 2006], mas também é denominado na literatura de *limpeza de dados*[2] [Chaudhuri et al., 2003], *pareamento de registros*[3] [Bhattacharya & Getoor, 2004; Fellegi & Sunter, 1969; Koudas et al., 2006] e *casamento de registros*[4] [Verykios et al., 2003]. Mais especificamente, a deduplicação de registros em repositórios de dados consiste na identificação e remoção de registros que se referem ao mesmo objeto ou entidade do mundo real, ainda que apresentem estilos de escrita, grafias, tipos de dados ou esquemas diferentes.

1.1 Motivação

Ultimamente, tem havido um grande investimento por parte de empresas e instituições governamentais no desenvolvimento de métodos eficientes para remoção de réplicas em grandes repositórios de dados [Bell & Dravis, 2006; Wheatley, 2004]. Entretanto, a deduplicação de registros é uma tarefa bastante complexa, cujo tratamento requer muito tempo e poder de processamento devido à grande quantidade de comparações de registros necessárias. Logo, os métodos propostos para deduplicação de registros devem procurar atingir seus objetivos da forma mais eficiente possível.

Recentemente, de Carvalho et al. [2008a] apresentaram uma abordagem inovadora para a identificação de registros duplicados em repositórios de dados, recorrendo a uma técnica de aprendizado de máquina conhecida como *Programação Genética* (PG) [Banzhaf et al., 1998; Koza, 1992]. Através dessa abordagem, registros são deduplicados utilizando-se evidências extraídas do conteúdo dos dados para criar funções de similaridade, genericamente denominadas de *funções de deduplicação*, capazes de apontar quais registros do repositório são réplicas.

Entretanto, apesar dos resultados superiores a outras abordagens encontradas na literatura, técnicas baseadas em aprendizado de máquina geralmente necessitam de uma etapa de treino, na qual os exemplos para aprendizado dos padrões de duplicação

[2]Do inglês *data cleaning*.
[3]Do inglês *record linkage*.
[4]Do inglês *record matching*.

normalmente são gerados de forma manual. Dessa forma, o custo e tempo necessários para a criação do conjunto de exemplos de treino muitas vezes dificultam a utilização prática dessas técnicas.

1.2 Contribuição

O objetivo desta dissertação é propor uma abordagem baseada em uma técnica determinística que seja capaz de sugerir, de forma automática, exemplos para a etapa de treino do processo de deduplicação de registros utilizando PG.

Inicialmente, verificou-se a real necessidade de se utilizar todos os pares de exemplos gerados para a etapa de treino. Foram realizados diversos experimentos nos quais a quantidade desses pares de exemplos foi reduzida gradualmente, verificando-se como cada redução afetava a qualidade e o desempenho do processo de geração das funções de deduplicação para a tarefa de deduplicação de registros. Em seguida, um método determinístico foi utilizado para a geração dos exemplos de treino para o processo de deduplicação de registros utilizando PG, permitindo uma análise da viabilidade de se selecionar exemplos de forma automática. Por fim, foram apontadas as principais características dos repositórios de dados que facilitam (e dificultam) a utilização da abordagem para seleção automática de exemplos de treino para deduplicação de registros.

Assim sendo, a principal contribuição desta dissertação é uma abordagem para seleção automática de exemplos de treino para um método de deduplicação de registros utilizando PG [Gonçalves et al., 2009]. Resultados experimentais mostram que é possível utilizar uma quantidade reduzida de exemplos de treino sem afetar a qualidade das soluções obtidas ao final do processo de geração das funções de deduplicação, reduzindo de forma significativa o tempo necessário para a execução da etapa de treino.

1.3 Trabalhos Relacionados

A deduplicação de registros é um tópico de pesquisa que tem atraído bastante interesse em bancos de dados e áreas relacionadas. Como já ressaltado, a ocorrência de réplicas em repositórios de dados leva a inconsistências que podem afetar severamente diversos tipos de serviço, causando prejuízos para empresas e instituições governamentais.

Na tentativa de resolver essas inconsistências, alguns trabalhos propõem a criação de funções de similaridade capazes de combinar as informações contidas nos repositórios de dados para identificar quando um par de registros constitui ou não uma réplica. Elmagarmid et al. [2007] classificam essas abordagens em duas categorias:

***Ad-Hoc*:** são abordagens que geralmente dependem do conhecimento de um domínio ou de métricas de distância específicas (por exemplo, para cadeias de caracteres).

Baseadas em Treino: são abordagens que dependem de algum tipo de treinamento, supervisionado ou semi-supervisionado, para a identificação de réplicas, como abordagens probabilísticas e de aprendizado de máquina.

Alguns trabalhos importantes são apresentados a seguir.

1.3.1 Abordagens *Ad-hoc*

Em [Chaudhuri et al., 2003] é proposto um algoritmo de pareamento que recebe um registro de um arquivo (ou repositório de dados) e procura por outro registro em um arquivo de referência que "case" com o primeiro, de acordo com alguma função de similaridade pré-definida. Os registros pareados são selecionados de acordo com um limiar de similaridade mínima definido pelo usuário, permitindo que mais de um registro candidato seja retornado como resposta. Nestes casos, o usuário fica responsável por escolher o registro duplicado que mais se aproxima do registro original.

Um método de pesos é utilizado no WHIRL [Cohen, 2000], um sistema gerenciador de bancos de dados que suporta junções por similaridade entre relações que apresentam atributos textuais. Os pesos são calculados pelo conhecido método TF-IDF [Baeza-Yates & Ribeiro-Neto, 1999].

1.3.2 Abordagens baseadas em Treino

Por estarem mais relacionadas ao trabalho realizado nesta dissertação, as abordagens baseadas em treino são apresentadas mais detalhadamente em duas sub-seções: abordagens probabilísticas e abordagens baseadas em aprendizado de máquina.

1.3.2.1 Abordagens Probabilísticas

Newcombe et al. [1959] foram os primeiros a tratar o problema de deduplicação de registros como um problema de inferência Bayesiana (um problema probabilístico), propondo uma abordagem para lidar com o problema de forma automática. Entretanto,

Elmagarmid et al. [2007] fazem uma crítica a essa abordagem ao afirmar que, apesar de inovadora, a mesma não apresenta uma base estatística sólida.

Fellegi & Sunter [1969] formalizaram a intuição do trabalho de Newcombe et al. [1959] e propuseram uma elaborada abordagem estatística para lidar com o problema de combinação de evidências. O método proposto requer a definição de dois limiares para identificação de réplicas. Se o valor de similaridade entre duas entidades estiver acima do *limiar de identificação positiva*, elas são consideradas réplicas; se estiver abaixo do *limiar de identificação negativa*, as mesmas são considerados não-réplicas; e se o valor de similaridade estiver entre os dois limiares, as entidades são classificadas como "possíveis réplicas", exigindo que a classificação seja feita por um especialista.

Se por um lado o método de Fellegi & Sunter [1969] não exige que o usuário forneça exemplos para a realização de uma etapa de treino, como acontece com as abordagens de aprendizado de máquina, ele tem a desvantagem de exigir que o usuário defina manualmente os dois limiares de identificação citados acima, uma tarefa que geralmente não é trivial, uma vez que esses valores dependem de características do repositório de dados que será deduplicado. Essa abordagem dominou a área por mais de duas décadas, até que fossem desenvolvidas novas técnicas de deduplicação pelas comunidades de aprendizado de máquina e estatística. O Febrl[5] [Christen, 2008] é uma das ferramentas que implementam esse método.

1.3.2.2 Abordagens baseadas em Aprendizado de Máquina

As técnicas de aprendizado de máquina, por sua vez, necessitam de uma porção de dados para treino. Esses dados devem apresentar as mesmas características do conjunto de dados a ser deduplicado, tornando possível uma generalização das soluções obtidas para o restante do repositório de dados original. O maior problema desse tipo de abordagem é o custo de criação do conjunto de treino, uma tarefa que pode ser pouco viável em muitos casos.

Tejada et al. [2001] apresentam um sistema chamado *Active Atlas*, cujo objetivo principal é aprender regras para mapear registros a partir de dois arquivos distintos, estabelecendo relacionamentos entre eles. Durante a etapa de aprendizado, são definidos os pesos de transformação e as regras de mapeamento. O processo de combinação dos pesos é executado utilizando árvores de decisão.

Já em [Tejada et al., 2002], é apresentada uma estratégia baseada em aprendizado ativo em que, novamente, árvores de decisão são utilizadas no ensino de regras para a deduplicação de registros com múltiplos atributos. O método sugere que, com a

[5]Freely Extensible Biomedical Record Linkage – http://sourceforge.net/projects/febrl

criação de múltiplos classificadores, treinados com dados ou parâmetros ligeiramente diferentes, é possível detectar casos ambíguos e então pedir uma resposta por parte do usuário. Segundo Elmagarmid et al. [2007], a principal inovação desse trabalho está na criação de diversas funções redundantes e na exploração concorrente de suas ações conflitantes, visando a descoberta de novos tipos de inconsistência entre réplicas no conjunto de dados.

Em [Bilenko et al., 2003] e [Bilenko & Mooney, 2003], os autores apresentam o MARLIN (*Multiply Adaptive Record Linkage with INduction*), um sistema que utiliza uma técnica de aprendizado de máquina para melhorar as funções de similaridade utilizadas na comparação de atributos dos registros e a forma como as evidências, vetores de termos utilizados para o treino de um classificador baseado em SVM (*Support Vector Machines*) [Joachims, 2002], são combinadas. Esse sistema ainda utiliza diversas estratégias de blocagem para aumentar a eficiência dos agrupamentos dos pares de registros similares.

Ainda em [Bilenko et al., 2003], os autores realizam uma comparação entre diferentes métricas de similaridade baseadas em símbolos (*tokens*) e caracteres, mostrando, por exemplo, que a métrica proposta pelos autores (SoftTF.IDF) apresenta os melhores resultados dentre as métricas avaliadas. Os autores também deixam bem claro que nenhuma métrica de similaridade é adequada para todos os tipos de repositório de dados, ou seja, métricas que demonstram robustez e bom desempenho para alguns repositórios de dados podem apresentar um desempenho abaixo do esperado em outros. Dessa forma, eles defendem a utilização de métricas mais flexíveis, capazes de suportar múltiplas comparações de similaridade, como em [Bilenko et al., 2003] e [Tejada et al., 2002].

Em [de Carvalho et al., 2006], os autores apresentam uma abordagem baseada em programação genética para melhorar os resultados do método de Fellegi & Sunter [1969], utilizando essa técnica de aprendizado de máquina para a geração de combinações de evidências melhores do que o simples somatório linear utilizado pelo método probabilístico.

Já em [de Carvalho et al., 2008a], é proposta uma nova abordagem baseada em PG para encontrar a melhor combinação de evidências em um arcabouço genérico independente de qualquer outra técnica. Uma vez que a identificação de réplicas é uma tarefa que consome muito tempo, mesmo para repositórios de dados pequenos, o método proposto tenta combinar os melhores fragmentos de evidências para a geração de funções de similaridade que maximizem o desempenho, utilizando para isto uma pequena porção do repositório de dados para treino.

Por fim, de Carvalho et al. [2008b] apresentam um detalhado estudo experimental para mostrar como a seleção dos parâmetros do processo de PG pode afetar o desempenho do método de geração das funções de deduplicação de registros, sugerindo que a escolha de valores mais adequados para os parâmetros conduzem o processo a soluções mais rápidas e eficientes. A principal contribuição desse trabalho é um conjunto de instruções para a definição dos parâmetros de controle da programação genética para o problema de deduplicação de registros, uma vez que reduz-se o esforço para a definição dos parâmetros mais adequados para o problema, além de fornecer explicações detalhadas de cada um dos parâmetros e do impacto de cada um deles sobre os resultados finais. Esses parâmetros de controle da programação genética são discutidos na Seção 2.1.2.

Nesta dissertação, são apresentados resultados de um estudo experimental que mostra como o tamanho do conjunto de treino do processo de deduplicação de registros utilizando PG influencia a qualidade das soluções obtidas ao final do processo de geração das funções de deduplicação. Além disso, um método determinístico é utilizado para a geração do conjunto de treino para o processo de deduplicação de registros, permitindo realizar uma análise da viabilidade de se selecionar os exemplos para treino de forma automática e eliminando a necessidade de se gerar manualmente esses exemplos. Por fim, um projeto fatorial [Jain, 1991] é realizado para mostrar quais as características dos repositórios de dados que facilitam e dificultam a aplicação da abordagem proposta para seleção automática de exemplos de treino para o método de deduplicação de registros baseado em PG.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte forma. O Capítulo 2 faz uma introdução da técnica de programação genética e apresenta uma visão geral do processo de deduplicação de registros utilizando PG. O Capítulo 3, por sua vez, descreve a abordagem proposta para seleção automática de exemplos de treino para a deduplicação de registros baseada em PG e apresenta os resultados de uma série de experimentos realizados para a validação dessa abordagem. Já no Capítulo 4 é apresentado um extenso projeto fatorial realizado para identificar as características dos repositórios de dados que facilitam e dificultam a utilização da abordagem proposta no capítulo anterior. Finalmente, o Capítulo 5 descreve as conclusões finais deste trabalho e apresenta alguns possíveis trabalhos futuros.

Capítulo 2

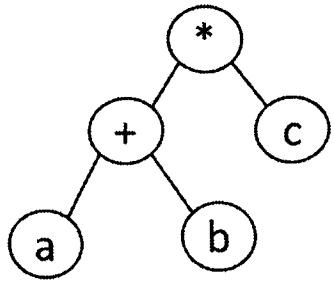
Deduplicação de Registros baseada em Programação Genética

Neste capítulo, é feita uma introdução à programação genética [Banzhaf et al., 1998; Koza, 1992], uma conhecida técnica de aprendizado de máquina utilizada neste trabalho. Serão apresentados os conceitos mais importantes, como as operações genéticas realizadas durante o processo, os principais parâmetros de controle e o algoritmo que descreve o processo evolucionário. Em seguida, esta técnica é discutida no contexto do problema de deduplicação de registros, sendo apresentada a abordagem proposta por de Carvalho et al. [2008a,b] e que serve como arcabouço para o trabalho desenvolvido nesta dissertação.

2.1 Programação Genética

A programação genética (PG) é uma das mais conhecidas e utilizadas técnicas de computação evolucionária, podendo ser vista como uma heurística adaptativa cujas idéias básicas são originadas do processo de seleção natural. É uma evolução direta dos programas ou algoritmos usados para aprendizado indutivo (supervisionado), inicialmente aplicados em problemas de otimização.

Uma das principais características das técnicas evolucionárias é a sua capacidade de tratar problemas com múltiplos objetivos, normalmente modelados como restrições do ambiente durante o processo evolucionário [Banzhaf et al., 1998]. Essas técnicas também são conhecidas pela capacidade de procurar por soluções em grandes – e possivelmente infinitos – espaços de busca, nos quais a solução ótima pode ser desconhecida, fornecendo geralmente respostas bem próximas do ótimo [Banzhaf et al., 1998; Koza, 1992].



árvore(a,b,c) = (a + b) * c

Figura 2.1. Exemplo de uma Função de Deduplicação mapeada como Árvore.

Um dos principais aspectos que diferencia PG das demais técnicas evolucionárias (como algoritmos genéticos e sistemas evolucionários) é a forma como representa os conceitos e interpreta o problema – como um programa de computador, sendo que os dados são vistos e manipulados desta forma. Programas de computador possuem a flexibilidade necessária para expressar soluções de uma grande variedade de problemas. Além disso, as estruturas dos programas em evolução não apresentam limitações de tamanho, podendo variar dinamicamente durante o processo, de acordo com as demandas do problema [Koza, 1992].

As representações mais utilizadas em PG são árvores e grafos. Neste trabalho, assim como em [de Carvalho et al., 2008a,b], foi utilizada uma representação baseada em árvores para a construção das funções de deduplicação – exemplificadas na Figura 2.1 – e para a representação das soluções para o problema.

Além de escolher uma representação para as soluções do problema, é necessário definir um conjunto de terminais e de funções para a realização da tarefa de geração das funções de deduplicação. Os terminais são entradas, constantes ou nós com aridade zero, também denominados folhas, enquanto o conjunto de funções é constituído por operadores, declarações e funções básicas ou definidas pelo usuário, utilizados durante o processo evolucionário para manipular os valores dos terminais [Koza, 1992]. Os nós folhas se encontram ao final dos ramos das árvores, enquanto as funções são colocadas em seus nós internos, como pode ser visto também na Figura 2.1. O espaço de busca é o espaço de todos os programas de computador formados pelas funções e terminais especificados no domínio do problema.

Uma vez que a representação de árvore foi escolhida para representar os indivíduos (possíveis soluções para o problema), é muito importante que, após a aplicação de cada operação genética, as árvores resultantes ainda sejam árvores válidas. Para isso,

as árvores são manipuladas por operações capazes de evitar situações que poderiam afetar a integridade da função global. Por exemplo, um nó folha nunca é substituído por um nó interno e vice-versa [Banzhaf et al., 1998]. As operações genéticas são explicadas na próxima seção.

2.1.1 Operações Genéticas

Durante o processo evolucionário, os *indivíduos* – possíveis soluções para o problema – são manipulados e modificados por operações genéticas como reprodução, cruzamento (*crossover*) e mutação, em um processo iterativo que tenta gerar indivíduos cada vez melhores a cada geração subsequente. Essas operações são explicadas a seguir.

A operação de reprodução consiste em copiar os indivíduos sem realizar qualquer tipo de modificação em suas estruturas. Geralmente, esta operação é utilizada para implementar uma estratégia elitista, mantendo o código genético dos indivíduos mais aptos inalterados no decorrer das gerações [Koza, 1992]. Dessa forma, se um bom indivíduo é encontrado nas gerações iniciais, dificilmente ele será perdido durante o processo evolucionário, após diversas aplicações de operações genéticas.

A operação de cruzamento se baseia na troca de conteúdo genético entre dois indivíduos pais, resultando em dois indivíduos filhos. Intuitivamente, se dois indivíduos são pelo menos um pouco efetivos na resolução do problema, então alguma de suas partes provavelmente possui algum mérito. Ao recombinar os fragmentos de alguns bons indivíduos, espera-se que sejam gerados indivíduos ainda melhores que seus pais, capazes de resolver o problema com maior êxito. Esta operação, exemplificada na Figura 2.2, pode ser descrita da seguinte forma:

Passo 1. Seleciona-se dois indivíduos (árvores pais) de acordo com alguma política de pareamento.

Passo 2. Escolhe-se aleatoriamente um fragmento de cada indivíduo (sub-árvore).

Passo 3. Permuta-se os dois fragmentos escolhidos.

Passo 4. Reinicia-se o processo evolucionário com os indivíduos resultantes do cruzamento (árvores filhas).

Durante a operação de cruzamento, todos os nós das árvores pais apresentam a mesma probabilidade de serem escolhidos, visando a manutenção da diversidade dos indivíduos dentro da população.

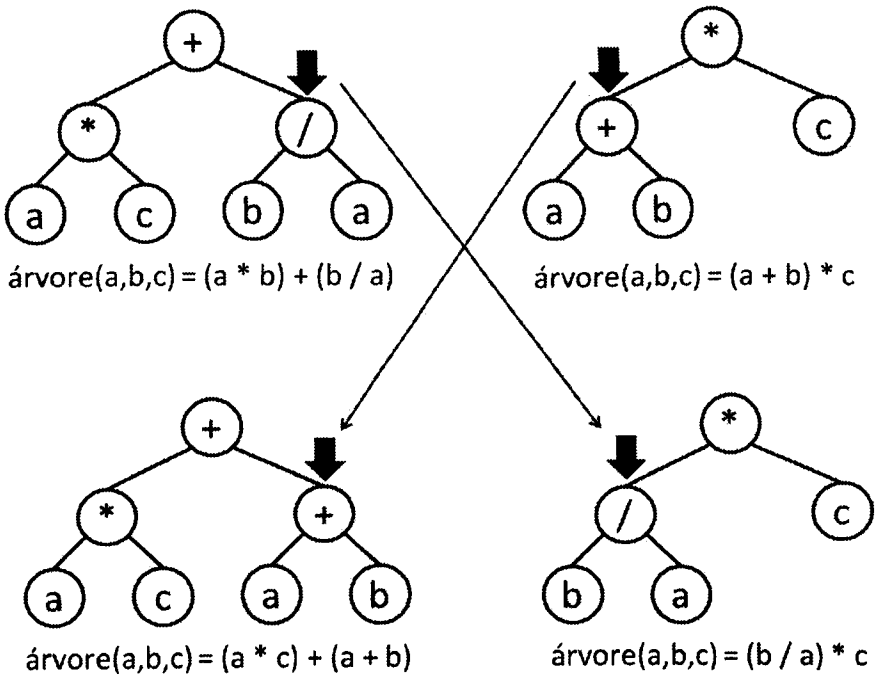


Figura 2.2. Exemplo de Execução da Operação de Cruzamento.

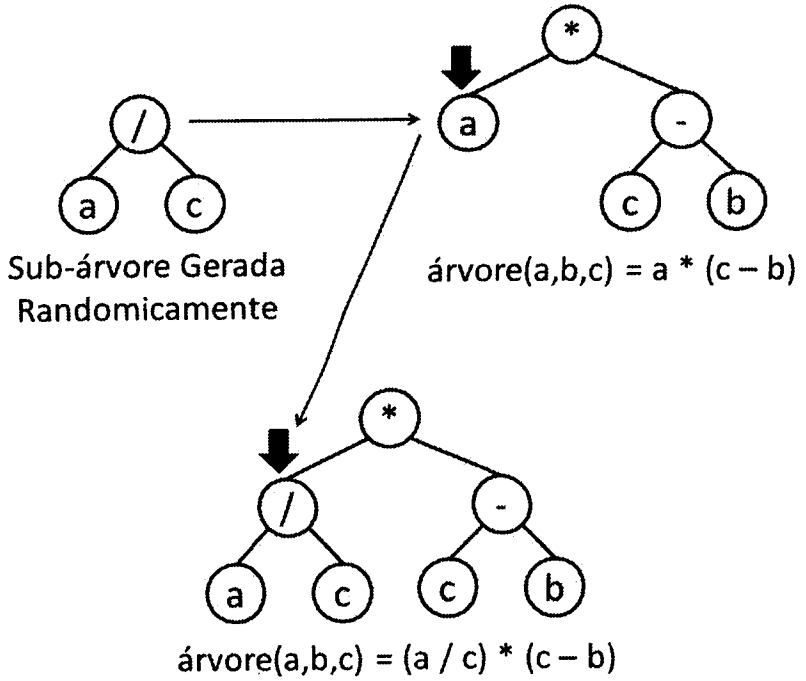


Figura 2.3. Exemplo de Execução da Operação de Mutação.

Já a operação de mutação, exemplificada na Figura 2.3, visa a manutenção de um nível mínimo de diversidade dos indivíduos na população, ajudando a programação genética a obter boas soluções mais rapidamente. O processo é descrito da seguinte forma:

Passo 1. Seleciona-se um indivíduo da população atual[1].

Passo 2. Seleciona-se aleatoriamente um fragmento desse indivíduo.

Passo 3. Cria-se aleatoriamente uma árvore de mutação.

Passo 4. Gera-se um novo indivíduo através da substituição do fragmento do indivíduo selecionado pela árvore de mutação criada no passo anterior.

O cruzamento e a mutação, por serem responsáveis pela modificação de soluções candidatas em novas soluções candidatas para o problema, são denominadas operações de transformação [Banzhaf et al., 1998].

2.1.2 Parâmetros de Controle

O paradigma de programação genética é controlado por diversos parâmetros numéricos (n) e qualitativos (q). Os principais parâmetros para o problema de deduplicação de registros são apresentados a seguir, conforme definidos em [de Carvalho et al., 2008b].

Número de Gerações (n) é a quantidade de ciclos (iterações) que serão executados durante o processo evolucionário da programação genética. Geralmente, quanto maior o número de gerações, melhores os resultados obtidos ao final do processo. Entretanto, este aumento exige mais tempo para a realização da etapa de treino.

Número de Execuções (n) é a quantidade de vezes que o processo evolucionário será executado em sequência. Quanto maior o número de execuções, melhores serão os resultados, visto que a qualidade dos indivíduos gerados é medida pela média dos resultados em cada execução, tornando-os mais confiáveis. Mais informação pode ser encontrada na Seção 4.2.

Método de Geração da População Inicial (q) é o método escolhido para a geração da primeira população do processo evolucionário, aquela que constituirá a geração 0 (zero). Os três métodos mais utilizados são: *FullDepth*, *Grow* e *Ramped Half-and-Half*. Utilizando o primeiro método, todas as árvores são criadas com

[1] Cada árvore resultante da etapa de cruzamento possui chances iguais de sofrer mutação.

o tamanho máximo, conforme especificado pelo parâmetro de controle *Tamanho Máximo da Árvore ou Indivíduo*. Ao utilizar o segundo método, as árvores são criadas com uma quantidade aleatória de nós, mas sem exceder a profundidade máxima pré-definida pelo usuário. Por fim, quando se utiliza o terceiro método, metade das árvores é criada usando o primeiro método, enquanto a outra metade é criada utilizando o segundo método.

Tamanho da População (n) se refere ao número de indivíduos (possíveis soluções para o problema) que serão processados em cada geração do processo evolucionário.

Tamanho Máximo da Árvore ou Indivíduo (n) é a maior altura que um indivíduo pode apresentar durante qualquer momento do processo evolucionário.

Tamanho Máximo da Árvore de Mutação (n) é o maior tamanho com que uma árvore de mutação pode ser criada durante a operação de mutação. Este valor deve ser menor ou igual ao *Tamanho Máximo da Árvore ou Indivíduo*.

Taxa de Mutação (n) é a probabilidade de um indivíduo da população sofrer mutação. Mais informação sobre a operação de mutação pode ser encontrada na Seção 2.1.1.

Método de Seleção dos Indivíduos (q) é o método utilizado para selecionar os indivíduos mais aptos, aqueles que seguirão para a geração seguinte. Os métodos mais comuns são: *Roulette Wheel* (a probabilidade de seleção é proporcional à aptidão do indivíduo), *Tournament* (para cada posição disponível na próxima geração, uma quantidade predefinida de indivíduos é selecionada aleatoriamente e o mais apto é escolhido), *Random* (a seleção é feita de forma aleatória), *Ranking* (as n posições disponíveis na geração seguinte são ocupadas pelos n indivíduos mais aptos) e *Greedy* (um pequeno grupo formado pelos indivíduos mais aptos possui maiores chances de serem selecionados do que o restante dos indivíduos). Mais informações sobre aptidão dos indivíduos são encontradas na Seção 2.2.

Método de Pareamento dos Indivíduos (q) é o método usado para parear os indivíduos durante a operação de cruzamento (*crossover*). Os mais utilizados são: *Random*, *Ranking* e *Mirror*. O primeiro consiste em escolher os indivíduos que serão pareados de forma aleatória, como o próprio nome diz. No segundo, os indivíduos são pareados em ordem crescente de aptidão. Já no terceiro método, os indivíduos são ordenados pelo valor de aptidão e pareados da seguinte forma: *(1 n), (2 n-1), (3 n-2), ..., ((n/2) (n/2)-1)*, sendo n o tamanho da população.

Em [de Carvalho et al., 2008b], os autores apresentam os resultados de um estudo experimental que mostra como a escolha dos valores desses parâmetros de controle influencia no desempenho da tarefa de geração das funções de deduplicação, no que diz respeito aos valores de F1 médio e desvio padrão dos melhores indivíduos gerados ao final desse processo. Ao utilizar (os denominados) bons valores de parâmetros, mostrou-se que a diferença na qualidade dos indivíduos pode chegar a 30%, deixando claro que a escolha de valores ruins faz com que a tarefa de geração de funções de deduplicação de registros exija mais tempo e recursos do que o necessário para a sua realização.

Como veremos nos Capítulos 3 e 4, neste trabalho, os valores utilizados para esses parâmetros, na ordem apresentada acima, foram: *30*, *10* (para os experimentos do Capítulo 3) e *30* (para os experimentos do Capítulo 4), *Ramped Half-and-Half*, *50*, *5*, *4*, *2%*, *Ranking* e *Random*.

2.1.3 Algoritmo Evolucionário Geracional

Neste trabalho, o processo evolucionário é guiado por um algoritmo evolucionário geracional, com ciclos de gerações distintos e bem definidos. Essa abordagem foi utilizada por conseguir capturar a idéia básica por trás dos algoritmos evolucionários. Após a execução de várias gerações, espera-se que a população contenha indivíduos mais aptos, capazes de solucionar o problema da melhor forma possível.

O algoritmo do processo evolucionário é descrito – em alto-nível – da seguinte forma:

Passo 1. A população inicial é gerada (aleatoriamente ou utilizando indivíduos gerados manualmente).

Passo 2. Cada indivíduo da população atual é *avaliado* e recebe um valor numérico que mede a sua aptidão individual.

Passo 3. Se o critério de parada é alcançado, o Passo 7 é executado e o algoritmo termina. Caso contrário, a execução do algoritmo continua.

Passo 4. *Reproduz* os n melhores indivíduos para a população da geração seguinte.

Passo 5. Utilizando um processo de *seleção*, são escolhidos m indivíduos que farão parte da geração seguinte.

Passo 6. As operações genéticas de *cruzamento* e *mutação* são aplicadas a todos os indivíduos selecionados, direcionando seus descendentes para a população seguinte.

Por fim, a população gerada substitui a população existente e o algoritmo retorna ao Passo 2.

Passo 7. O(s) melhor(es) indivíduo(s) da população é(são) apresentado(s) como resultado do processo evolucionário.

Neste trabalho, a população inicial foi gerada sempre de forma aleatória (Passo 1).

A avaliação realizada no Passo 2 é uma operação que consiste na atribuição de um valor que representa a capacidade de cada indivíduo gerado durante o processo evolucionário de lidar com o problema em questão. Neste trabalho, os indivíduos são avaliados de acordo com sua capacidade de prever boas soluções para o problema de identificação de réplicas, utilizando os conjuntos de funções e terminais disponíveis. Esse valor resultante é denominado aptidão do indivíduo (*individual fitness*), enquanto as funções de avaliação são denominadas funções de aptidão (*fitness functions*). A natureza das funções de aptidão varia de acordo com o problema. Para definir essas funções, é essencial que se tenha um bom conhecimento do domínio do problema a ser resolvido, uma vez que, se forem mal escolhidas, o processo dificilmente obterá boas soluções.

Para determinar o final da evolução, ou seja, atingir o critério de parada citado no Passo 3 do algoritmo evolucionário descrito acima, o usuário pode fixar o número de gerações que serão executadas, o tempo máximo de processamento ou condicionar o algoritmo à obtenção de uma solução satisfatória, atingindo um ponto considerado ótimo. Neste trabalho, o número máximo de gerações executadas durante o processo evolucionário foi utilizado como condição de parada.

Já o processo de seleção citado no Passo 5 é responsável pela escolha dos indivíduos que seguirão para a próxima geração do processo evolucionário, de acordo com algum critério pré-definido. Este operador utiliza os valores de aptidão gerados durante o processo de avaliação para decidir quais são os indivíduos mais aptos. Estratégias para o processo de seleção podem utilizar desde técnicas muito simples, como apenas escolher n indivíduos quaisquer (técnica conhecida como *random selection*), a técnicas mais complexas, como ordenar decrescentemente os indivíduos, atribuir probabilidades de acordo com os valores de aptidão e selecioná-los aleatoriamente de acordo com estas probabilidades (técnica conhecida como *roullete wheel*). Mais informações sobre programação genética podem ser encontradas em [Banzhaf et al., 1998] e [Koza, 1992].

2.2 Visão Geral da Abordagem para Deduplicação de Registros baseada em Programação Genética

Para realizar a deduplicação de registros, são utilizadas funções que combinam evidências, sendo que cada evidência E é formada por um par *<atributo, função de similaridade>* que representa o uso de uma função de similaridade específica sobre valores de um determinado atributo do repositório de dados.

Por exemplo, para deduplicar a tabela de um banco de dados relacional com os atributos *nome*, *sobrenome*, *idade* e *endereço*, utilizando a função de similaridade Jaro-Winkler (JW) [Winkler, 1999], teríamos a seguinte lista de evidências:

E_1<*nome*, JW>, E_2<*sobrenome*, JW>, E_3<*idade*, JW> e E_4<*endereço*, JW>.

Para este exemplo, uma função simples (F_s) poderia ser uma combinação linear da forma

$$F_s(E_1, E_2, E_3, E_4) = E_1 + E_2 + E_3 + E_4, \tag{2.1}$$

enquanto uma função mais complexa (F_c) poderia ser da forma

$$F_c(E_1, E_2, E_3, E_4) = E_1 \times (\frac{E_2}{E_3^{E_4}}). \tag{2.2}$$

Para modelar as funções em formato de árvore, cada evidência é representada por uma folha, através de valores reais normalizados entre 0,0 e 1,0, enquanto os nós internos representam as operações aritméticas (por exemplo, $+$, $-$, $\times$, $\div$, exp) que manipulam os valores das folhas.

Conforme explicado na Seção 2.1.1, durante o processo evolucionário, os indivíduos são manipulados e modificados através de diversas operações genéticas, em um processo que tenta gerar indivíduos melhores em cada geração subsequente. Todas as árvores geradas durante este processo são avaliadas automaticamente, ou seja, cada possível solução para o problema é testada em repositórios de dados – com características semelhantes às do conjunto de treino – onde as réplicas já foram previamente identificadas. Além de permitir a avaliação da capacidade de identificar pares de registros que sejam réplicas verdadeiras, esta automatização viabiliza o uso dessa técnica.

As entradas das funções são formadas por instâncias de evidências extraídas dos dados manipulados. Já a saída consiste no resultado da operação codificada em cada árvore, valor que é comparado com um limiar de identificação de réplicas da seguinte forma: se o valor for superior ao limiar, os registros são considerados réplicas; caso

contrário, os registros são considerados distintos. Essa abordagem de classificação obedece as propriedades de transitividade das réplicas, de forma que, se um registro A for réplica de um registro B e B for réplica de um registro C, então A será réplica de C.

Experimentos realizados em de Carvalho et al. [2008a] mostram que a abordagem para deduplicação de registros baseada em PG consegue adaptar as funções de deduplicação geradas de acordo com mudanças no limiar de identificação de réplicas, necessário para classificar os pares de registros. Dessa forma, o usuário não precisa se preocupar em definir valores para esse limiar de acordo com o repositório de dados, visto que as funções de deduplicação sugeridas pelo método se ajustam automaticamente, mantendo o nível de qualidade das soluções, apesar das mudanças no valor do limiar.

Após uma comparação entre todos os pares de registros gerados, contabiliza-se o número total de identificações de réplicas corretas e incorretas. Essa informação é utilizada posteriormente pela função de aptidão, componente responsável pela avaliação dos indivíduos gerados durante todo o processo evolucionário. A métrica *F1* foi escolhida como função de aptidão para os experimentos deste trabalho. Ela combina harmonicamente as tradicionais métricas de *precisão* e *revocação* utilizadas em avaliações de sistemas de recuperação de informação [Baeza-Yates & Ribeiro-Neto, 1999; Bilenko et al., 2003] da seguinte forma:

$$Precisão = \frac{QuantidadeDeParesDeRéplicasIdentificadosCorretamente}{QuantidadeDeParesDeRéplicasIdentificados} \tag{2.3}$$

$$Revocação = \frac{QuantidadeDeParesDeRéplicasIdentificadosCorretamente}{QuantidadeDeParesDeRéplicasExistentes} \tag{2.4}$$

$$F1 = \frac{2 \times Precisão \times Revocação}{Precisão + Revocação} \tag{2.5}$$

A precisão é responsável por mensurar a proporção de réplicas identificadas corretamente dentre todas as identificações realizadas, ou seja, de todos os pares de registros que foram identificados como réplicas pela função de deduplicação, quantos pares realmente são réplicas. Já a revocação é utilizada para calcular a proporção de réplicas identificadas corretamente dentre todas as identificações que deveriam ter sido feitas, ou seja, de todos os pares de registros que deveriam ter sido identificados como réplicas, quantos foram devidamente identificados.

Uma vez que a precisão e a revocação são métricas relacionadas, capazes de capturar diferentes aspectos da identificação de réplicas no contexto de deduplicação de registros, decidiu-se pela utilização de uma única métrica capaz de combinar precisão e

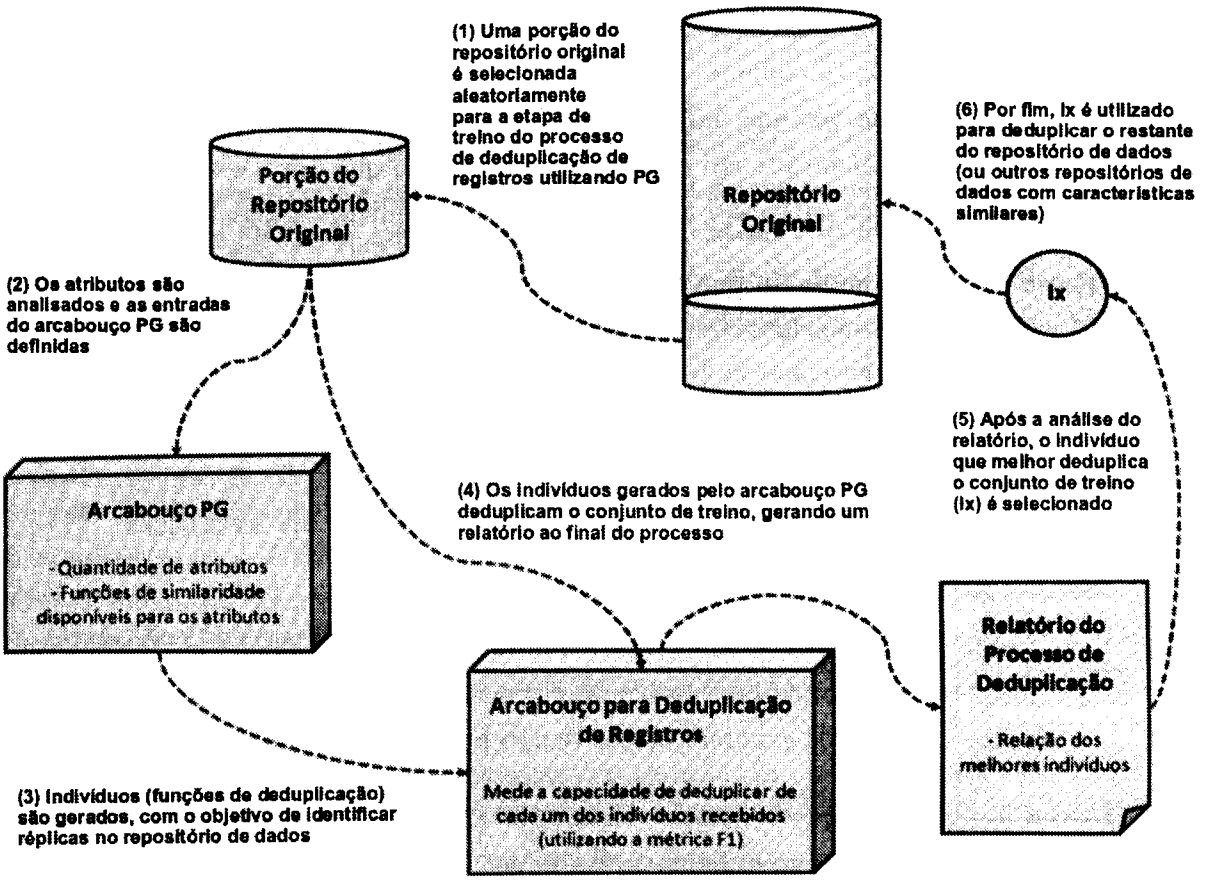


Figura 2.4. Esquema do Processo de Deduplicação de Registros utilizando a Abordagem baseada em Programação Genética.

revocação [Baeza-Yates & Ribeiro-Neto, 1999]. Dessa forma, a métrica F1 foi utilizada para representar, através de um único valor (entre 0 e 1), o quão bem um indivíduo consegue identificar réplicas em um repositório de dados. É importante ressaltar que a métrica F1 assume valores elevados apenas quando os valores de precisão e revocação também forem elevados.

2.2.1 Deduplicação de Registros utilizando a Abordagem baseada em PG

A Figura 2.4 apresenta uma visão geral do processo de deduplicação de registros utilizando a abordagem baseada em programação genética proposta por de Carvalho et al. [2008a,b]. Uma descrição detalhada de cada etapa é apresentada a seguir.

Etapa 1. Uma porção do repositório de dados a ser deduplicado é selecionada aleatoriamente para a etapa de treino. Nos experimentos realizados nos trabalhos citados anteriormente, o conjunto de treino corresponde a 25% do repositório de dados original, uma vez que, no início do processo de deduplicação, este repositório é dividido igualmente em quatro arquivos, sendo que um deles é utilizado para treino e os demais para a avaliação (teste) dos indivíduos gerados.

Etapa 2. Os atributos do repositório de dados são analisados e as entradas do arcabouço PG são definidas, ou seja, verifica-se o tipo de cada um dos atributos para então selecionar as funções de similaridade mais adequadas para cada um deles. Dessa forma, o arcabouço PG mantém uma lista com os atributos e as respectivas funções de similaridades utilizadas para a criação das evidências. Para a realização dos experimentos deste trabalho, as evidências foram criadas utilizando as funções de similaridade Jaro [Koudas et al., 2006] para os atributos de texto e Distância de Edição [Koudas et al., 2006] para os atributos numéricos. Segundo experimentos preliminares realizados em [de Carvalho et al., 2008a], estas funções de similaridade se mostraram as mais adequadas para os respectivos tipos de dados, além de exigirem menos tempo para o processamento das evidências.

Etapa 3. Ao final de cada geração do processo evolucionário, são selecionados indivíduos (possíveis soluções para o problema) com o objetivo de se identificar registros réplicas em um determinado repositório de dados.

Etapa 4. No arcabouço para deduplicação de registros, os indivíduos gerados são utilizados para identificar réplicas na porção do repositório de dados utilizada para teste, gerando um relatório ao término do processo evolucionário. Neste relatório, é apresentada a relação dos melhores indivíduos de cada geração e os seus respectivos valores de aptidão, medidos pela métrica F1.

Etapa 5. Após uma análise do relatório do processo de deduplicação, seleciona-se para o passo seguinte o melhor indivíduo obtido, ou seja, a função que melhor conseguiu deduplicar o conjunto de dados de treino.

Etapa 6. O melhor indivíduo obtido é então utilizado para deduplicar o restante do repositório de dados, podendo também ser utilizado para a deduplicação de outros repositórios com características semelhantes.

Uma visível desvantagem do processo descrito acima pode ser vista no primeiro passo. Nos trabalhos citados no início desta seção, o conjunto de treino é formado

sempre por 25% do repositório de dados original, quantidade esta que pode exigir um tempo de treino muito superior ao realmente necessário, já que uma grande quantidade de registros deve ser comparada, atributo por atributo.

No Capítulo 3, será apresentada uma abordagem para seleção automática dos exemplos de treino. Após a realização de uma série de experimentos, apresentados ao final daquele capítulo, mostra-se que é possível selecionar – de forma automática – uma porção bastante reduzida de exemplos para treino, diminuindo consideravelmente o tempo necessário para a realização da etapa de treino e mantendo um nível satisfatório de qualidade dos indivíduos gerados.

2.2.2 Análise de Complexidade do Processo de Geração das Funções de Deduplicação

Nesta seção, é apresentada uma aproximação da análise de complexidade do processo de geração das funções de deduplicação, visto que existem métodos mais sofisticados e específicos para verificar a complexidade de problemas que utilizam programação genética.

A complexidade de tempo do processo de geração das funções de deduplicação é dada por $O(N_{ger} \times N_{ind} \times T_{aval})$, onde N_{ger} é o número de gerações utilizadas para a evolução dos indivíduos, N_{ind} é o número de indivíduos (possíveis soluções para o problema) que constituem uma população e T_{aval} é a complexidade do processo de avaliação da aptidão (*fitness*) de cada indivíduo. O fator T_{aval}, por sua vez, possui complexidade $O(N_{tam}^2)$, onde N_{tam} é o tamanho do repositório, dado pelo número de registros a serem deduplicados. No pior caso, a identificação das réplicas requer uma comparação entre todos os registros do repositório de dados de treino, o que explica o expoente quadrático da complexidade de T_{aval}. Logo, a complexidade do processo de geração das funções de deduplicação é dada por $O(N_{ger} \times N_{ind} \times N_{tam}^2)$.

Capítulo 3

Abordagem para Seleção Automática de Exemplos de Treino

Neste capítulo, apresentamos a abordagem proposta para a seleção automática de exemplos de treino do processo de deduplicação de registros utilizando programação genética. Alguns conceitos relevantes para o entendimento do processo serão abordados inicialmente, seguidos por uma explicação mais detalhada sobre o funcionamento da abordagem proposta. Por fim, serão apresentados os resultados do estudo experimental utilizando a abordagem proposta, juntamente com uma validação da mesma.

3.1 Definição de Pares de Registros

No início do processo de deduplicação de registros utilizando programação genética, os registros são agrupados em pares, de forma que possam ser comparados através de funções de similaridade predefinidas pelo usuário, com a intenção de se apontar quais registros do repositório são realmente réplicas. Dessa forma, ao falar sobre exemplos de treino, é necessário introduzir os conceitos de *pares de registros positivos* e *pares de registros negativos*.

Um par de registros positivo é formado por dois registros que fazem referência ao mesmo objeto ou entidade do mundo real, ou seja, após compará-los utilizando uma função de similaridade predefinida, eles são apontados como réplicas um do outro. Por outro lado, um par de registros negativo é constituído por dois registros que não fazem referência ao mesmo objeto do mundo real, não sendo apontados como réplicas um do outro. Esses dois tipos de exemplo de treino auxiliam o método de deduplicação de registros utilizando PG a compreender o que é e o que não é réplica dentro de um repositório de dados.

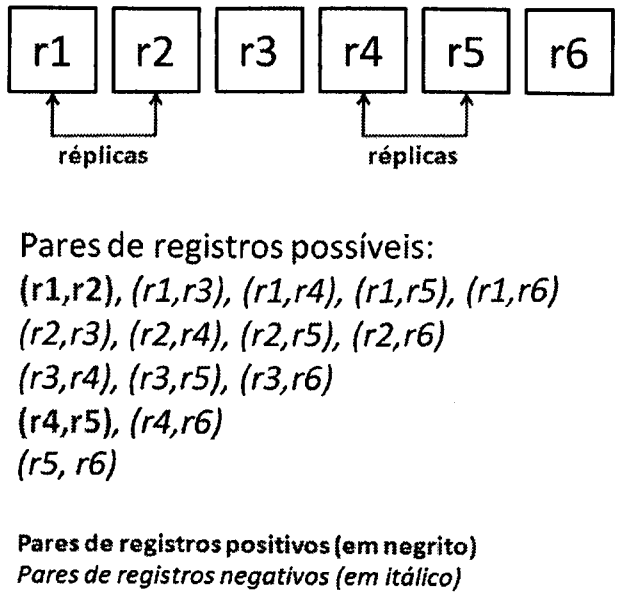


Figura 3.1. Exemplo de Pares de Registros Positivos e Pares de Registros Negativos em um Repositório de Dados composto por Seis Registros.

Para se identificar réplicas em um repositório de dados, cada registro deve ser comparado com todos os demais. Logo, se determinado repositório contém n registros, devem ser realizadas $\frac{n \times (n-1)}{2}$ comparações entre dois registros. O divisor é igual a dois pois, na realidade, apenas metade das comparações é realizada, já que dois registros nunca são comparados mais de uma vez. A maioria das comparações corresponderão a não-réplicas (pares de registros negativos), uma vez que o número máximo de pares duplicados é geralmente menor que a quantidade de registros no repositório [Christen, 2008]. A Figura 3.1 exemplifica pares de registros positivos e pares de registros negativos em um repositório de dados composto por seis registros.

3.2 Seleção Automática de Exemplos de Treino

Em [de Carvalho et al., 2008a,b], a geração das funções de deduplicação exige que o usuário selecione uma porção do repositório de dados para a etapa de treino e identifique manualmente cada um dos pares de registros positivos e negativos neste conjunto. Ao lidar com repositórios de dados pequenos, essa tarefa pode até ser realizada sem muita dificuldade, dependendo da quantidade de "sujeira" do repositório e do nível de conhecimento do usuário sobre o mesmo. Entretanto, quando o usuário precisa lidar com repositórios de dados maiores e mais complexos, a geração manual dos exemplos de treino se torna extremamente custosa e inviável.

Outro problema com relação à geração manual dos exemplos de treino diz respeito ao tamanho do conjunto de treino. Definir a quantidade de pares de exemplos (positivos e negativos) que permita a geração de boas funções de deduplicação, ou seja, que sejam capazes de identificar a maior quantidade possível das réplicas existentes no repositório de dados, no menor tempo possível, também não é uma tarefa trivial.

Para evitar que o usuário tenha que criar manualmente o conjunto de treino para a geração das funções de deduplicação, a abordagem proposta neste capítulo utiliza uma técnica determinística para selecionar automaticamente um subconjunto desses exemplos para utilização na etapa de treino (Etapa 4 do processo apresentado na Seção 2.2.1). Os passos envolvidos ao se aplicar essa abordagem são apresentados a seguir.

Passo 1. O repositório de dados a ser deduplicado é dividido igualmente em quatro arquivo, sendo um deles para treino e os demais para avaliação. Uma das principais diferenças entre esta abordagem e a Etapa 1 do processo apresentado na Seção 2.2.1 está na quantidade de registros utilizados na etapa de treino: ao invés de se utilizar todos os pares de registros gerados no arquivo para treino, será utilizado apenas uma porção deles. Os experimentos apresentados mais adiante, na Seção 3.3, mostram a relação entre a quantidade de pares de registros utilizados para treino e os resultados obtidos ao final do processo de geração das funções de deduplicação, no que diz respeito ao tempo gasto para se realizar a etapa de treino e à qualidade dos indivíduos gerados.

Passo 2. O método determinístico de Fellegi & Sunter [1969], descrito na Seção 1.3, deduplica o conjunto de treino e gera duas listas: uma com todos os pares de registros positivos e outra com todos os pares de registros negativos. Essa deduplicação é realizada utilizando-se a ferramenta Febrl [Christen, 2008]. Os limiares (inferior e superior) para identificação de réplicas são definidos sempre com o mesmo valor[1]. Esta simplificação evita a necessidade de se identificar manualmente os pares de registros que ficariam no intervalo entre os dois limiares (pares de registros cuja identificação é mais difícil). As funções de similaridade utilizadas para a construção das evidências são as mesmas utilizadas pelo método de deduplicação de registros utilizando PG, apresentado na Seção 2.2.1.

Passo 3. São definidos os valores percentuais de pares de registros positivos e pares de registros negativos a serem utilizados na etapa de treino. As devidas quantidades

[1]Valores definidos após a realização de experimentos iniciais.

de pares de registros são extraídas das duas listas de exemplos geradas no passo anterior, resultando em um novo conjunto de exemplos para treino.

Passo 4. Por fim, executa-se normalmente o processo de deduplicação de registros utilizando PG, mas a partir da Etapa 2 descrita na Seção 2.2.1, usando como entrada o conjunto de exemplos de treino gerado no passo anterior.

Dessa forma, a criação do conjunto de exemplos de treino fica mais fácil, viabilizando a utilização prática da técnica de PG para lidar com o problema de deduplicação de registros, uma vez que se elimina a necessidade de qualquer interação humana no processo de seleção dos exemplos para treino. É importante ressaltar que, embora o método utilizado no Passo 2 da abordagem descrita tenha sido o de Fellegi & Sunter [1969], é possível utilizar outros métodos determinísticos de classificação, como o k-means [Gu & Baxter, 2006].

3.3 Resultados Experimentais

Nesta seção, são apresentados os resultados dos experimentos realizados para mostrar como a abordagem proposta para seleção automática de exemplos de treino afeta a qualidade das soluções geradas e o tempo gasto para a execução da etapa de treino do processo de deduplicação de registros utilizando PG.

Na primeira parte do estudo experimental, foram realizados três conjuntos de experimentos, variando-se percentualmente e de forma gradual a quantidade de pares de registros (positivos e negativos) utilizados na etapa de treino. Primeiramente, reduziu-se a quantidade de pares de registros positivos, enquanto os pares de registros negativos foram mantidos em sua totalidade. Em seguida, a mesma redução foi feita para os pares de registros negativos, mantendo-se a totalidade dos pares de registros positivos. Por fim, tanto a quantidade de pares de registros positivos quanto de pares de registros negativos foram reduzidas. Os valores dos percentuais de redução foram escolhidos de forma empírica, após a realização de experimentos iniciais.

O objetivo desses experimentos é mostrar como a escolha da quantidade de exemplos (pares de registros positivos e negativos) utilizados na etapa de treino afeta o desempenho do processo de geração das funções de deduplicação de registros utilizando PG. Além disso, procurou-se verificar se é realmente necessário utilizar todos os exemplos gerados para treino, como foi feito em [de Carvalho et al., 2008a,b]. Os resultados desta avaliação permitirão a sugestão de configurações para a seleção de exemplos de treino, possibilitando a identificação de réplicas em repositórios de dados de forma mais eficiente, mas sem prejudicar a qualidade das soluções geradas.

Na segunda parte do estudo experimental, utilizou-se a abordagem proposta para seleção de exemplos de treino para a realização de um novo conjunto de experimentos. Desta vez, o objetivo foi avaliar, através da métrica F1, se os exemplos gerados automaticamente por um método determinístico se mostraram bons exemplos de treino. Esta seleção automática torna o processo de deduplicação de registros utilizando técnicas de aprendizagem de máquina mais acessível.

Ao final dos experimentos, são apresentados os valores de F1 médio e desvio padrão do melhor indivíduo em cada arquivo de teste, após 10 execuções, conforme realizado em [de Carvalho et al., 2008b]. A configuração dos parâmetros da PG, conforme apresentado na Seção 2.1.2, é a seguinte: *30* gerações, *10* execuções (*30* para os experimentos do Capítulo 4), método *Ramped Half-and-Half* para a geração da população inicial, população formada por *50* indivíduos, tamanho máximo da árvore igual a *5*, tamanho máximo da árvore de mutação igual a *4*, taxa de mutação igual a *2%*, método *Ranking* para a seleção dos indivíduos e o método *Random* para o pareamento dos indivíduos. As evidências (pares <atributo, função de similaridade>) utilizadas neste trabalho são as mesmas estabelecidas para experimentação em [de Carvalho et al., 2008a].

Todos os experimentos foram realizados utilizando computadores com a seguinte configuração: processador Pentium Core 2 Quad de 2 GHz, com 4 GB RAM DDR2 de memória principal e HD SATA de 320 GB, rodando o sistema operacional FreeBSD[2] 7.1 64-Bits e utilizando a linguagem de programação Python[3] na versão 2.5.1.

3.3.1 Conjunto de Dados Experimental

Para a criação dos conjuntos de dados necessários para experimentação, utilizou-se o SDG [Christen, 2005], um gerador de dados sintéticos disponível na ferramenta Febrl [Christen, 2008]. Este gerador permite a criação de conjuntos de dados contendo nomes (baseados em tabelas de frequência para nomes e sobrenomes), endereços (baseados em tabelas de frequência para localidades, códigos postais, números de ruas, etc.), números telefônicos e identificadores numéricos para pessoas (número de seguridade social).

Uma vez que dados reais não são facilmente disponibilizados para experimentação, devido a restrições de privacidade e confidencialidade, a utilização de dados sintéticos foi considerada a melhor opção. Além disso, ela permite uma melhor avaliação do impacto na qualidade das soluções finais, resultado das mudanças na quantidade de pares de registros positivos e negativos utilizada na etapa de treino, uma vez que os erros existentes e as características dos registros são conhecidos *a priori*.

[2]http://www.freebsd.org/
[3]http://www.python.org

Os dados gerados pelo SDG são similares aos frequentemente encontrados em registros de dados médicos pessoais. Primeiramente, a ferramenta cria um conjunto de dados contendo apenas registros originais. Em seguida, réplicas são geradas a partir desses registros por meio de modificações como inserção, remoção e substituição de caracteres, além de troca, remoção, inserção e divisão de palavras, alterações que são baseadas em características reais de erros. As réplicas são então inseridas no conjunto de dados original para a realização dos experimentos de deduplicação de registros. Cada registro é formado pelos seguintes atributos: *nome*, *sobrenome*, *número da rua*, *endereço1*, *endereço2*, *bairro*, *código postal*, *estado*, *data de aniversário*, *idade*, *número telefônico* e *número de seguridade social*.

Para experimentação, foi criado um repositório de dados sintéticos contendo 2.000 registros, distribuídos igualmente em quatro arquivos, sendo cada um deles composto por 300 registros originais e 200 réplicas. Essas réplicas foram geradas obedecendo as seguintes restrições: no máximo uma réplica pôde ser criada baseada em um registro original (utilizando uma distribuição uniforme), no máximo uma modificação pôde ser feita em cada atributo do registro e no máximo um atributo pôde ser alterado no registro todo. Uma vez que o arquivo de treino é formado por 500 registros, será gerado um total de 124.750 pares de registros (vide cálculo apresentado ao final da Seção 3.1), sendo 200 pares de registros positivos e 124.550 pares de registros negativos.

Nos experimentos apresentados neste capítulo, utilizou-se uma proporção maior de registros replicados nos conjuntos de dados de treino e de teste do que a encontrada em cenários reais, como o reportado pelo projeto USIIS[4], em que a taxa de registros replicados é de aproximadamente 20%. Apesar disso, apenas exemplos representativos, ou seja, aqueles mais úteis para o aprendizado dos padrões de duplicação, são efetivamente utilizados no treino [de Carvalho et al., 2008a].

3.3.2 Descrição dos Experimentos

Conforme explicado anteriormente, o objetivo dos experimentos apresentados nesta seção é observar como a qualidade dos indivíduos gerados no processo de deduplicação e o tempo gasto para treino são afetados pelas mudanças na quantidade de exemplos utilizados na etapa de treino do processo de deduplicação de registros utilizando PG. Em todos os experimentos realizados, os pares de registros foram selecionados de forma aleatória, nas proporções definidas para cada experimento.

[4]Utah Statewide Immunization Information System – http://health.utah.gov/phi/brownbag/handouts/2008/USIIS_april.pdf

Tabela 3.1. Redução Percentual de Pares de Registros Positivos – Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Negativos	% Pares Positivos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
100	100	37,97	0,994	0,009	0,997	0,008	0,995	0,011
	95	41,00	0,996	0,004	1,000	0,000	0,997	0,003
	90	43,37	0,997	0,003	1,000	0,000	0,997	0,003
	70	36,80	0,996	0,006	1,000	0,000	0,998	0,001
	50	41,20	0,996	0,002	1,000	0,000	1,000	0,000
	30	40,90	0,995	0,003	1,000	0,000	0,999	0,001
	15	41,42	0,994	0,004	0,998	0,004	0,998	0,004
	5	42,52	0,988	0,010	0,991	0,010	0,989	0,014
	2,5	33,00	0,967	0,035	0,972	0,032	0,967	0,040
	1	32,32	0,952	0,052	0,957	0,049	0,949	0,055
	0	39,62	0,315	0,418	0,317	0,414	0,321	0,419

3.3.2.1 Experimentos com Redução Percentual de Pares de Registros Positivos

Neste conjunto de experimentos, a quantidade de pares de registros positivos utilizados no treino foi reduzida gradualmente, enquanto os pares de registros negativos foram mantidos em sua totalidade. Apesar de serem minoria no conjunto de exemplos utilizados nessa etapa, os pares de registros positivos afetam a qualidade dos indivíduos gerados no processo de deduplicação. Os resultados deste conjunto de experimentos são apresentados na Tabela 3.1.

O tempo total gasto – em horas – na etapa de treino de cada experimento deste conjunto também é apresentado na Tabela 3.1, sendo útil para efeito de comparação. Os resultados obtidos na etapa de teste são apresentados na quarta coluna em diante, indicando o valor médio de F1 e o desvio padrão do melhor indivíduo em cada arquivo de teste (A, B e C). Os resultados dos demais conjuntos de experimentos são apresentados em tabelas com a mesma estrutura.

Os resultados mostram que a redução da quantidade de pares de registros positivos utilizados na etapa de treino afeta a qualidade (medida pela métrica F1) dos resultados obtidos na etapa de teste. Entretanto, ao utilizar percentuais reduzidos de pares de registros positivos (por exemplo, 5 e 2,5%), ainda é possivel obter resultados próximos daqueles obtidos quando todos os pares de exemplos de treino são utilizados, onde os valores médios de F1 são elevados e os desvios padrões são reduzidos, mostrando que ocorre uma baixa dispersão dos valores de F1 em torno da média.

Ao utilizar apenas 2,5% dos pares de registros positivos, por exemplo, obtém-se uma economia no tempo de treino de aproximadamente 13%. Entretanto, não foi possível estabelecer uma relação direta entre o percentual de registros positivos e o tempo gasto para treino.

Tabela 3.2. Redução Percentual de Pares de Registros Negativos – Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Positivos	% Pares Negativos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
100	100	37,97	0,994	0,009	0,997	0,008	0,995	0,011
	95	44,13	0,998	0,002	0,999	0,003	0,999	0,001
	90	35,92	0,998	0,003	1,000	0,000	0,998	0,003
	70	26,83	0,998	0,002	0,999	0,003	0,998	0,003
	50	21,08	0,997	0,007	0,999	0,002	0,998	0,001
	30	12,02	0,993	0,009	0,999	0,003	0,994	0,011
	15	5,88	0,995	0,009	0,996	0,007	0,996	0,006
	5	2,33	0,926	0,138	0,925	0,132	0,905	0,185
	2,5	1,35	0,846	0,154	0,851	0,137	0,826	0,183
	1	0,60	0,677	0,242	0,649	0,260	0,654	0,266
	0	0,33	0,163	0,213	0,163	0,221	0,143	0,286

Quando os pares de registros positivos são completamente descartados, observa-se uma redução drástica nos valores de F1 médio e um aumento considerável nos valores de desvio padrão. Nessa situação, os indivíduos gerados identificam praticamente todos os pares de registros como réplicas, ou seja, eles conseguem identificar a maioria (ou mesmo a totalidade) dos pares de réplicas mas erram ao considerar muitos dos pares de registros negativos como positivos. Dessa forma, os valores de revocação ficam sempre próximos de 1,0 e os de precisão ficam próximos de 0,0. Como a F1 engloba as duas métricas, os valores de F1 médio dos melhores indivíduos serão sempre reduzidos, pois não basta apenas identificar todos os pares de réplicas. Os valores de F1 dos indivíduos são severamente penalizados pela elevada taxa de falsos positivos (pares não réplicas identificados como réplicas).

3.3.2.2 Experimentos com Redução Percentual de Pares de Registros Negativos

Já neste conjunto de experimentos, a quantidade de pares de registros do tipo mais frequente no conjunto de treino é que foi gradualmente reduzida. Por esse motivo, é importante analisar como essa redução influencia a qualidade dos indivíduos gerados no treino, além dos tempos gastos para a realização desta etapa.

Os resultados, apresentados na Tabela 3.2, demonstram que a redução da quantidade de pares de registros negativos influencia a qualidade dos resultados obtidos de forma mais significativa que a redução dos pares de registros positivos.

Na Tabela 3.1, por exemplo, observa-se que os valores médios de F1 para a configuração definida com 1% dos pares de registros positivos e a totalidade dos pares de registros negativos estão próximos de 0,950, enquanto a definição de 5% dos pares de registros negativos e a totalidade dos pares de registros positivos, vide Tabela 3.2, já leva a resultados inferiores aos obtidos utilizando a primeira configuração. Apesar disso, uti-

Tabela 3.3. Redução Percentual de Pares de Registros Positivos e Pares de Registros Negativos – Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Positivos	% Pares Negativos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
100	100	37,97	0,994	0,009	0,997	0,008	0,995	0,011
50	50	22,03	0,995	0,002	1,000	0,000	0,999	0,001
25	25	10,85	0,994	0,004	0,996	0,006	0,998	0,002
10	10	4,62	0,988	0,012	0,990	0,011	0,992	0,010
5	5	2,38	0,936	0,142	0,942	0,132	0,924	0,189
2,5	2,5	1,10	0,941	0,051	0,948	0,051	0,952	0,042
1	1	0,67	0,869	0,110	0,875	0,118	0,852	0,150

lizando uma quantidade reduzida de pares de registros negativos (por exemplo, 15%), já é possível obter soluções próximas daquelas utilizando todos os exemplos de treino. Neste caso, a economia no tempo de treino chega a ser de aproximadamente 85%.

Além disso, observa-se uma relação direta entre a quantidade de pares de registros negativos utilizados no treino e o tempo total gasto nesta etapa, ou seja, quanto menos pares de registros negativos são utilizados, menor é o tempo de treino, e vice-versa. Essa redução ocorre neste conjunto de experimentos uma vez que os pares de registros negativos correspondem à maioria dos exemplos utilizados na etapa de treino.

Neste repositório, ao se reduzir 85% dos pares de registros negativos, por exemplo, são desconsiderados 105.868 pares de registros para a etapa de treino, enquanto que a mesma redução percentual de pares de registros positivos corresponde à remoção de apenas 170 pares de registros. Dessa forma, obtém-se uma considerável economia no tempo de treino, visto que a quantidade de pares de registros utilizados nessa etapa é bastante reduzida.

3.3.2.3 Experimentos com Redução Percentual de Pares de Registros Positivos e Pares de Registros Negativos

Finalmente, neste conjunto de experimentos, são reduzidas as quantidades de pares de registros positivos e pares de registros negativos, gradualmente e na mesma proporção. Os resultados podem ser vistos na Tabela 3.3.

Novamente, observa-se uma relação direta entre a quantidade de pares de registros utilizados (positivos e negativos) e o tempo gasto na etapa de treino. Dessa forma, é possível reduzir consideravelmente a quantidade de pares utilizados para treino e ainda assim obter bons resultados. Utilizando 10% dos pares de registros positivos e 10% dos pares de registros negativos, por exemplo, a economia no tempo de treino chega a aproximadamente 88%, com perdas na qualidade da deduplicação de apenas 0,6%, 0,7% e 0,3%, nos arquivos de teste A, B e C, respectivamente.

Tabela 3.4. Utilizando o Método de Fellegi & Sunter [1969] para a Geração do Conjunto de Treino – Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Positivos	% Pares Negativos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
10	10	2,20	0,985	0,011	0,980	0,021	0,974	0,024
5	5	1,41	0,979	0,017	0,975	0,020	0,974	0,025
2,5	2,5	1,07	0,986	0,014	0,982	0,023	0,978	0,022

3.3.2.4 Deduplicação de Registros Determinística

A intenção deste último conjunto de experimentos é validar o processo de seleção automática de exemplos para a etapa de treino do processo de deduplicação de registros utilizando PG. Em todos os experimentos realizados, os pares de registros sugeridos pelo método determinístico foram selecionados de forma aleatória, nas proporções definidas para cada experimento. Após a realização dos experimentos anteriores, percebeu-se que é possível utilizar satisfatoriamente percentuais reduzidos de pares de registros positivos e negativos na etapa de treino. Dessa forma, os experimentos utilizaram apenas configurações com reduzidos percentuais de exemplos de treino. Os resultados podem ser vistos na Tabela 3.4.

Neste conjunto de experimentos, utilizando exemplos sugeridos automaticamente pelo método de Fellegi & Sunter [1969], os indivíduos gerados ao final do processo de deduplicação apresentaram elevados valores médios de F1, eliminando-se a necessidade de identificação manual dos exemplos de treino. Foi possível utilizar 2,5% dos pares de registros positivos e negativos para a etapa de treino sem que ocorresse uma redução considerável na qualidade dos indivíduos gerados ao final do processo de deduplicação de registros, uma vez que os valores médios de F1, comparando com a configuração que utiliza todos os pares de registros gerados, tiveram uma redução de apenas 0,8%, 1,5% e 1,7%, para os arquivos de teste A, B e C, respectivamente. Já a redução no tempo de treino foi superior a 97%.

Dessa forma, pode-se dizer que é possível utilizar satisfatoriamente a abordagem proposta para seleção automática de exemplos de treino para o processo de deduplicação de registros utilizando PG, pelo menos quando se utiliza repositórios de dados com características semelhantes aos que foram usados nos experimentos descritos neste capítulo. O Capítulo 4 apresenta um estudo que mostra que é possível utilizar um conjunto reduzido de treino para deduplicar repositórios de dados com os mais diversos níveis de dificuldade de deduplicação e ainda assim obter resultados satisfatórios.

Capítulo 4

Avaliação da Abordagem Proposta

Nos experimentos descritos no capítulo anterior, utilizamos um repositório de dados com uma determinada configuração. No entanto, é preciso verificar se a abordagem proposta para seleção automática de exemplos de treino consegue lidar com repositórios de dados de diferentes níveis de dificuldade, uma vez que, no mundo real, dois repositórios dificilmente apresentam as mesmas características e níveis de "sujeira".

A presença de uma grande quantidade de registros duplicados pode dificultar a identificação de réplicas em um repositório de dados, assim como a presença de registros que apresentem atributos com muitas variações ou que apresentem muitos atributos com variações. Geralmente, mensurar o grau de dificuldade de um repositório de dados não é uma tarefa trivial.

Neste capítulo, apresentamos os resultados de um projeto fatorial [Jain, 1991] que foi realizado para identificar quais as situações (características dos repositórios de dados) que facilitam ou dificultam a utilização da abordagem proposta para a seleção automática de exemplos de treino para o método de deduplicação de registros utilizando programação genética. Novamente, os repositórios de dados necessários para experimentação foram criados utilizando a ferramenta SDG [Christen, 2005].

Primeiramente, foram realizados alguns experimentos para determinar quais os percentuais ideais de pares de registros a serem utilizados para experimentação, visto que a realização de dezenas de experimentos – para diferentes percentuais de pares de registros – seria uma tarefa inviável. É importante ressaltar que a própria experimentação do projeto fatorial já exige muito tempo e esforço computacional, de modo que esses experimentos iniciais permitiram determinar os percentuais de registros relevantes para o projeto fatorial.

Tabela 4.1. Características dos Repositórios Utilizados para a Realização dos Experimentos Iniciais: Repositório MF ("Mais Fácil") e Repositório MD ("Mais Difícil").

Característica	Repositório MF	Repositório MD
Quantidade de registros originais a serem criados	**400**	**350**
Quantidade de réplicas a serem criadas	**100**	**150**
Número máximo de réplicas criadas a partir de um determinado registro original	**2**	**3**
Número máximo de modificações realizadas em cada atributo de uma réplica	**3**	**5**
Número máximo de atributos que podem ser modificados em uma réplica	**6**	**20**

4.1 Experimentos Iniciais

Inicialmente, foi feita uma caracterização de carga utilizando um repositório de dados com um nível reduzido de "sujeira" (Repositório MF), ou seja, réplicas não muito distintas dos respectivos registros originais estão presentes em pequenas quantidades neste repositório de dados. Em seguida, foi realizada outra caracterização de carga, mas desta vez utilizando um repositório de dados com um nível de "sujeira" mais elevado (Repositório MD): réplicas pouco semelhantes aos respectivos registros originais estão presentes em maiores quantidades neste repositório. As características dos Repositórios MF e MD são apresentadas na Tabela 4.1.

Para a geração das réplicas a partir dos registros originais, foi utilizada a distribuição de Poisson, tanto para o Repositório MF quanto para o Repositório MD. Essa distribuição foi escolhida por ser a que melhor mimetiza as probabilidades de erros encontrados em repositórios reais [Christen, 2005].

Os experimentos seguiram as mesmas configurações daqueles descritos no Capítulo 3, utilizando as mesmas reduções percentuais de pares de registros positivos e negativos. Analogamente, cada um dos repositórios foi dividido igualmente em quatro arquivos, sendo um deles utilizado para a etapa de treino e os demais para a etapa de teste. Os resultados dos experimentos utilizando o Repositório MF são apresentados na Tabela 4.2 e nas Figuras 4.1 e 4.3, enquanto os resultados dos experimentos utilizando o Repositório MD são apresentados na Tabela 4.3 e nas Figuras 4.2 e 4.4.

As Figuras 4.1 e 4.2 apresentam o tempo gasto (em horas) para a execução da etapa de treino do processo de deduplicação de registros utilizando PG para cada um dos experimentos realizados. Os resultados reforçam algumas conclusões obtidas ao final da experimentação descrita na Seção 3.3.2:

- A simples redução da quantidade de pares de registros positivos utilizados no treino, cujos experimentos correspondem à primeira curva exibida nas legendas dos gráficos apresentados nas Figuras 4.1 e 4.2, não apresenta uma relação direta com o tempo gasto para a etapa de treino do processo de deduplicação;

Tabela 4.2. Redução de Pares de Registros utilizando o Repositório MF ("Mais Fácil") - Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Negativos	% Pares Positivos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
100	*100*	*24,87*	*0,994*	*0,009*	*0,997*	*0,008*	*0,995*	*0,011*
	95	25,15	0,935	0,086	0,950	0,078	0,940	0,062
	90	23,73	0,900	0,112	0,920	0,100	0,915	0,079
	70	17,68	0,834	0,102	0,860	0,091	0,870	0,074
	50	19,50	0,891	0,104	0,912	0,094	0,906	0,072
100	30	22,50	0,913	0,070	0,933	0,068	0,899	0,088
	15	24,15	0,915	0,100	0,932	0,089	0,920	0,067
	5	17,65	0,811	0,097	0,833	0,104	0,765	0,116
	2,5	16,00	0,750	0,107	0,807	0,056	0,744	0,055
	1	18,22	0,553	0,113	0,638	0,148	0,529	0,110
	0	26,27	0,118	0,258	0,120	0,256	0,112	0,245
95		22,47	0,920	0,104	0,932	0,089	0,933	0,076
90		21,32	0,911	0,098	0,935	0,091	0,926	0,072
70		20,23	0,970	0,018	0,974	0,027	0,961	0,030
50		10,33	0,925	0,084	0,942	0,074	0,930	0,061
30	100	8,45	0,956	0,066	0,971	0,040	0,957	0,048
15		3,68	0,910	0,097	0,931	0,088	0,917	0,065
5		1,68	0,882	0,074	0,900	0,091	0,895	0,059
2,5		1,02	0,880	0,101	0,910	0,068	0,890	0,084
1		0,58	0,731	0,235	0,759	0,246	0,737	0,244
0		0,32	0,002	0,001	0,002	0,001	0,002	0,001
50	50	11,70	0,888	0,101	0,913	0,095	0,906	0,073
25	*25*	*5,80*	*0,875*	*0,119*	*0,917*	*0,084*	*0,878*	*0,115*
10	10	2,73	0,884	0,123	0,929	0,083	0,888	0,100
5	5	1,33	0,813	0,157	0,872	0,120	0,814	0,143
2,5	2,5	0,73	0,775	0,085	0,850	0,058	0,775	0,079
1	1	0,47	0,672	0,115	0,749	0,155	0,689	0,156

- A redução apenas da quantidade de pares de registros negativos e a redução da quantidade de pares de registros positivos e negativos, cujos experimentos correspondem à segunda e à terceira curvas, respectivamente, apresentam um comportamento diferente, contribuindo diretamente para a redução do tempo de treino do processo de deduplicação de registros.

Logo, um resultado mostrado inicialmente para apenas um repositório de dados, com um determinado grau de dificuldade de deduplicação, foi expandido para repositórios de dados com diferentes níveis de dificuldade.

Nas Figuras 4.3 e 4.4, cada gráfico apresenta os resultados dos experimentos em um determinado arquivo de teste (representados pelas letras A, B e C nas Tabelas 4.2 e 4.3), utilizando os Repositórios MF e MD, respectivamente. O *eixo x* representa os valores percentuais dos pares de registros que foram variados em cada conjunto de experimentos. O *eixo y*, por sua vez, representa os valores de F1 médio e desvio padrão do melhor indivíduo em cada um dos arquivos de teste. Seguindo a ordem apresentada nas legendas dos gráficos:

Tabela 4.3. Redução de Pares de Registros utilizando o Repositório MD ("Mais Difícil") - Tempos de Treino, Médias e Desvios Padrões de F1.

% Pares Negativos	% Pares Positivos	Tempo de treino (hs)	(A) Média	(A) Desvio Padrão	(B) Média	(B) Desvio Padrão	(C) Média	(C) Desvio Padrão
100	*100*	*20,57*	*0,822*	*0,109*	*0,804*	*0,120*	*0,812*	*0,092*
100	95	22,50	0,857	0,084	0,837	0,093	0,840	0,075
	90	26,02	0,858	0,110	0,840	0,122	0,842	0,091
	70	21,90	0,830	0,111	0,804	0,127	0,817	0,091
	50	21,85	0,832	0,077	0,804	0,101	0,807	0,071
	30	24,17	0,830	0,126	0,805	0,147	0,798	0,136
	15	18,27	0,723	0,162	0,686	0,188	0,672	0,175
	5	16,97	0,542	0,151	0,483	0,169	0,495	0,148
	2,5	20,28	0,663	0,163	0,642	0,202	0,619	0,206
	1	23,27	0,502	0,177	0,467	0,204	0,464	0,091
	0	25,47	0,062	0,099	0,071	0,111	0,066	0,107
95	100	20,82	0,807	0,122	0,790	0,143	0,801	0,103
90		21,93	0,839	0,099	0,842	0,099	0,830	0,095
70		16,78	0,808	0,114	0,794	0,133	0,802	0,092
50		13,42	0,880	0,074	0,872	0,069	0,856	0,073
30		6,78	0,839	0,102	0,825	0,115	0,819	0,084
15		4,02	0,851	0,066	0,832	0,056	0,816	0,050
5		1,42	0,693	0,117	0,672	0,132	0,659	0,135
2,5		1,07	0,736	0,069	0,730	0,079	0,682	0,068
1		0,58	0,466	0,172	0,457	0,170	0,410	0,155
0		0,35	0,004	0,000	0,004	0,000	0,004	0,000
50	50	13,55	0,838	0,102	0,829	0,118	0,827	0,082
25	*25*	*6,18*	*0,812*	*0,115*	*0,787*	*0,137*	*0,801*	*0,112*
10	10	1,92	0,699	0,132	0,697	0,118	0,707	0,090
5	5	1,15	0,567	0,207	0,630	0,163	0,594	0,159
2,5	2,5	0,85	0,689	0,217	0,700	0,175	0,688	0,178
1	1	0,57	0,602	0,138	0,598	0,124	0,612	0,109

A primeira curva (100P + xN) exibe os resultados dos experimentos onde apenas a quantidade de pares de registros negativos foi variada, mantendo a totalidade dos pares de registros positivos.

A segunda curva (100N + xP) apresenta os resultados dos experimentos onde apenas a quantidade de pares de registros positivos foi variada, mantendo a totalidade dos pares de registros negativos.

A terceira curva (xP + xN) representa os resultados dos experimentos onde tanto a quantidade de pares de registros positivos quanto a de pares de registros negativos foram variadas.

Analisando os resultados do Repositório MD, considerado mais difícil de deduplicar que o Repositório MF, pela Figura 4.4, nota-se que os valores de F1 médio ficaram próximos de 0,8 quando foram utilizados pelo menos 25% dos pares de registros positivos e 25% dos pares de registros negativos. Esta configuração permite a obtenção de bons resultados, com uma economia considerável no tempo de treino, como pode ser visto na Figura 4.2.

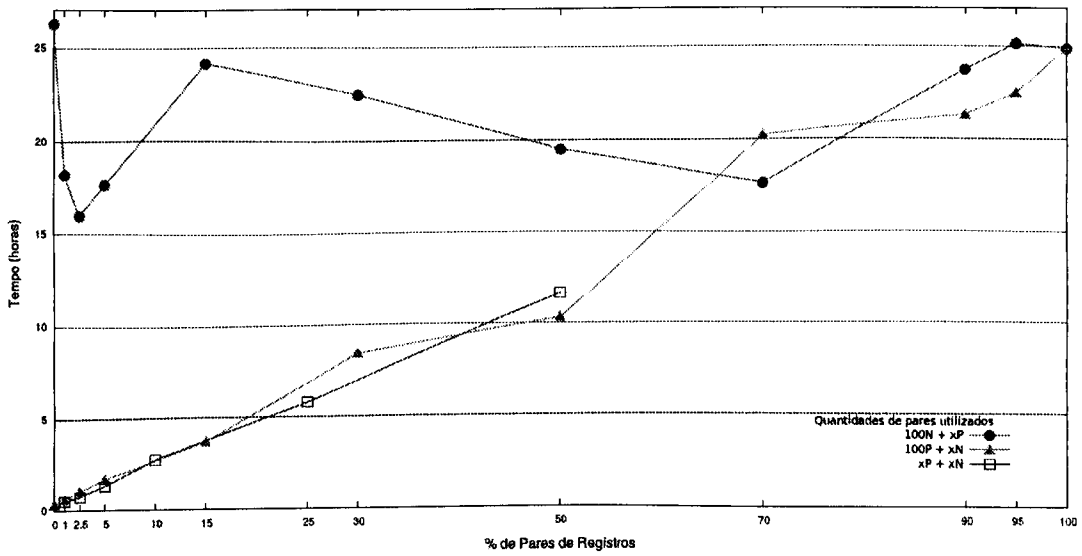


Figura 4.1. Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros para os Experimentos que utilizam o Repositório MF ("Mais Fácil").

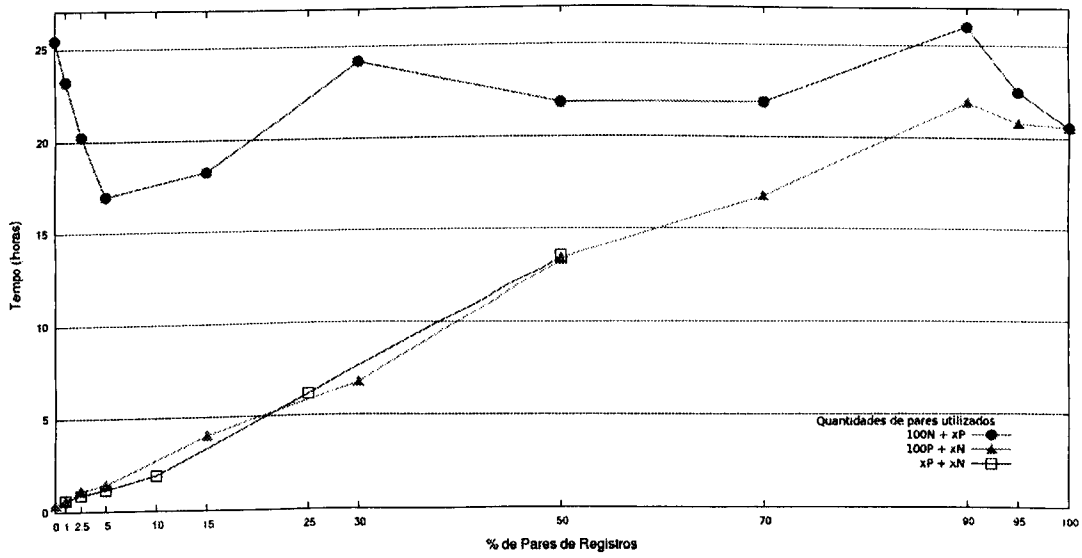


Figura 4.2. Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros para os Experimentos que utilizam o Repositório MD ("Mais Difícil").

Ao utilizar o Repositório MD, não houve perda na qualidade das soluções geradas, medida pelo F1 médio e desvio padrão do melhor indivíduo obtido ao final da geração das funções de deduplicação. Já ao utilizar o Repositório MF, notou-se pequenas perdas na qualidade das soluções, chegando a 6%, 3% e 7%, nos arquivos de teste A, B e C, respectivamente. Por outro lado, obteve-se uma economia no tempo de treino de aproximadamente 70% para o Repositório MD e de 77% para o Repositório MF.

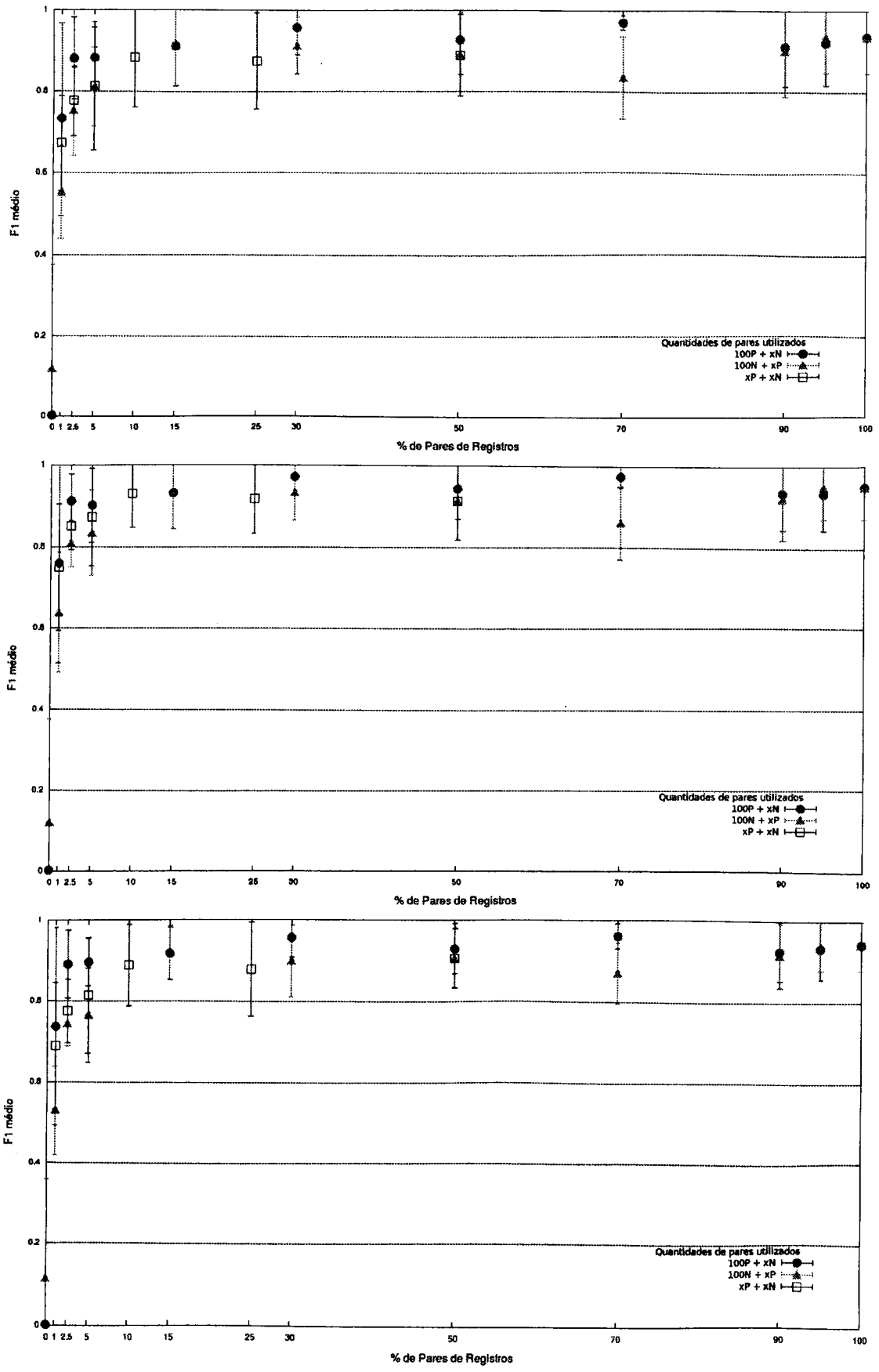


Figura 4.3. F1 Médio e Desvio Padrão do Melhor Indivíduo, nos Arquivos de Teste A, B e C, respectivamente, para os Experimentos que utilizam o Repositório MF ("Mais Fácil").

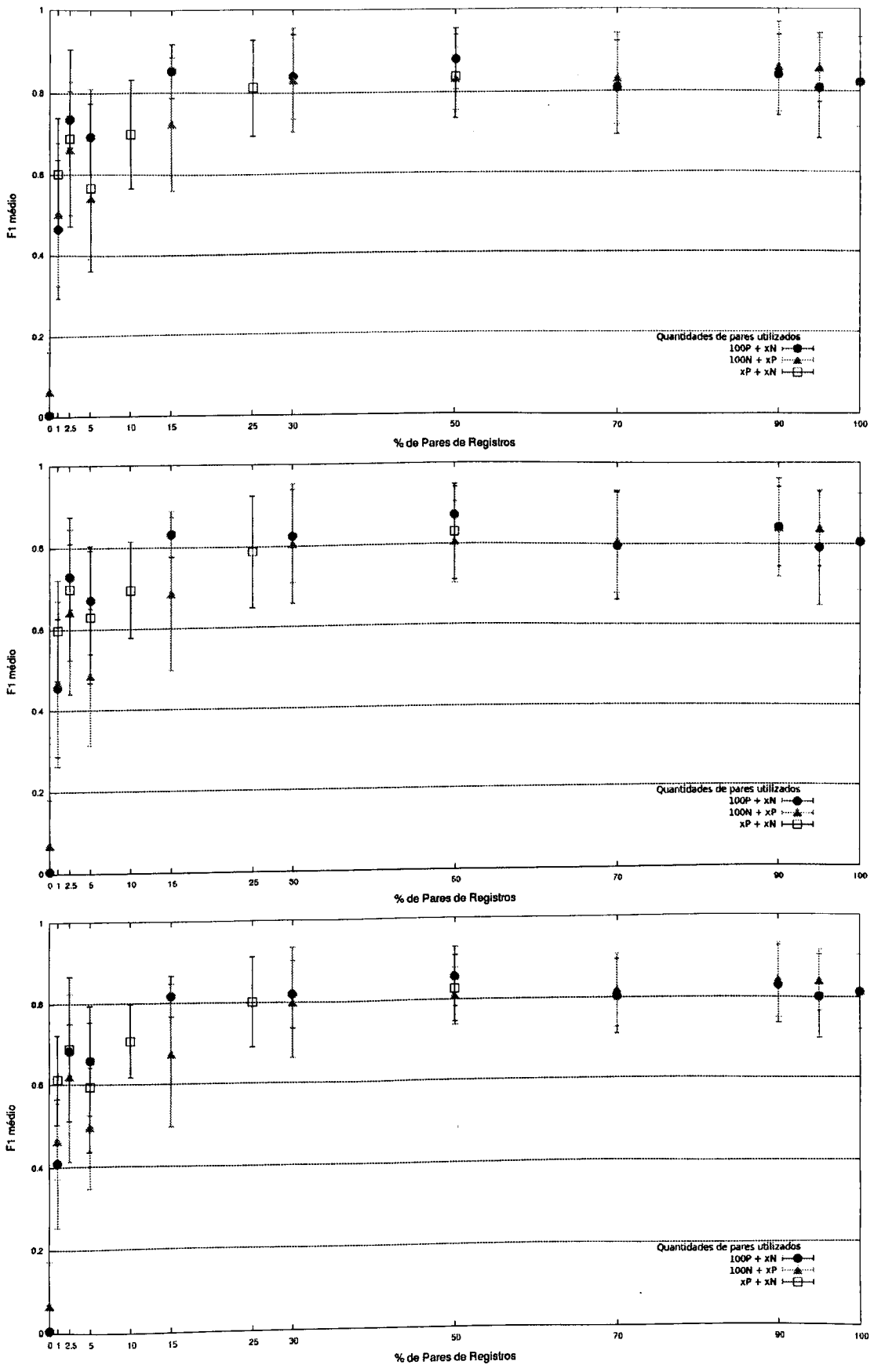


Figura 4.4. F1 Médio e Desvio Padrão do Melhor Indivíduo, nos Arquivos de Teste A, B e C, respectivamente, para os Experimentos que utilizam o Repositório MD ("Mais Difícil").

Repositórios de dados reais geralmente apresentam uma quantidade de registros superior à utilizada nos experimentos deste trabalho, o que evidencia ainda mais a importância de se obter tamanha economia nos tempos de treino, a etapa do processo de deduplicação de registros que consome mais tempo de execução. Não basta que o processo de deduplicação obtenha bons resultados, é preciso que ele seja eficiente.

Dessa forma, optou-se por utilizar duas configurações de pares de registros para os experimentos do projeto fatorial: uma composta pela totalidade dos pares de registros positivos e negativos e outra contendo 25% de cada um desses pares.

O objetivo desses experimentos iniciais foi mostrar que a seleção automática de exemplos de treino pode ser realizada com sucesso em repositórios de dados com diferentes graus de dificuldade de deduplicação, uma vez que, ao utilizar dois repositórios de dados com níveis de dificuldade tão díspares, os resultados obtidos foram bastante satisfatórios, possibilitando a obtenção de boas soluções com um tempo de treino bastante reduzido, viabilizando o processo de deduplicação de registros utilizando PG. Até então, os experimentos do Capítulo 3 haviam mostrado que a abordagem proposta para a seleção de exemplos de treino funcionava apenas para um repositório de dados específico. Além disso, esses experimentos forneceram um valor ideal para percentuais de exemplos de treino para a experimentação do projeto fatorial, descrito a seguir.

4.2 Projeto Fatorial

Na seção anterior, foram utilizados os termos "mais fácil" e "mais difícil" para caracterizar os repositórios de dados utilizados para experimentação. Entretanto, esses termos são muito vagos, devido à dificuldade de se mensurar o quanto o Repositório MF é "mais fácil" (de deduplicar) que o Repositório MD, ou mesmo quais são as características que o tornam mais fácil de ser deduplicado que o Repositório MD. Neste momento, surge a importância de se realizar um *Projeto Fatorial* [Jain, 1991].

O projeto fatorial é uma técnica utilizada para se obter o máximo de informação realizando a menor quantidade de experimentos possível, economizando tempo e esforço por parte do usuário. Uma análise minuciosa dos experimentos ajuda na quantificação dos efeitos causados pelos diversos fatores que podem afetar o desempenho do método utilizado. Dessa forma, é possível avaliar se determinado fator exerce algum efeito significativo nos resultados ou se as diferenças observadas ocorrem simplesmente devido às variações causadas pelos erros e parâmetros medidos e não controlados [Jain, 1991]. Alguns conceitos importantes são definidos a seguir:

Variável Resposta é o resultado do experimento. Geralmente, a variável resposta é medida pelo desempenho do sistema.

Fatores são as variáveis capazes de afetar a variável resposta.

Níveis são os valores que cada um dos fatores pode assumir (alternativas).

Replicação é a repetição de todos ou parte dos experimentos.

Projeto Experimental consiste na especificação da quantidade de experimentos, das combinações de níveis dos fatores para cada experimento e do número de replicações a serem realizadas para cada um dos experimentos.

Interação entre dois fatores ocorre se o efeito de um deles depende do nível do outro. Dessa forma, se um fator F_A é alterado do nível A_1 para o A_2 e sua perfomance é afetada de alguma forma, independendo do nível de um fator F_B, é possível afirmar que não há interação entre eles. Entretanto, se o fator F_A é alterado do nível A_1 para o A_2 e a performance varia de acordo com o nível do fator F_B, então existe uma interação entre os dois fatores.

Nesta dissertação, as variáveis resposta são o *F1 médio* e o *desvio padrão do melhor indivíduo* em cada arquivo de teste, juntamente com o *tempo gasto para treino* do processo de deduplicação de registros. Foram utilizados quatro parâmetros para criação dos repositórios de dados sintéticos como fatores do projeto fatorial: *DPO - Proporção de Registros Duplicados por Registro Original* (uma combinação do primeiro e segundo parâmetros para criação de repositórios de dados sintéticos, citados na primeira coluna da Tabela 4.1), *DBO - Número Máximo de Réplicas Criadas a partir de um Registro Original*, *MUA - Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica* e *MRT - Número Máximo de Modificações Realizadas em uma Réplica*.

A quantificação do impacto de cada um dos fatores analisados (e das interações entre eles) na qualidade dos indivíduos gerados ao final do processo de geração das funções de deduplicação de registros e no tempo gasto para treino é o principal objetivo deste projeto fatorial. Analisando os resultados, será possível identificar quais dos quatro fatores facilitam ou dificultam a utilização do método de deduplicação de registros utilizando PG e da abordagem de seleção de exemplos de treino.

Primeiramente, foram definidos os níveis (valores mínimos e máximos) para cada fator. Tais valores ajudam a quantificar a influência de cada fator e interação na qualidade das soluções geradas, ou seja, no F1 médio e no desvio padrão do melhor indivíduo obtido ao final do processo de geração das funções de deduplicação, em cada

Tabela 4.4. Níveis dos Fatores utilizados no Projeto Fatorial.

Fator / sigla	Valor Mínimo (-1)	Valor Máximo (+1)
Proporção de Registros Duplicados por Registro Original (DPO)	**0,25** 100 rép./400 orig.	**0,50** 167 rép./333 orig.
Número Máximo de Réplicas Criadas a partir de um Registro Original (DBO)	**2**	**4**
Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica (MUA)	**3**	**6**
Número Máximo de Modificações Realizadas em uma Réplica (MRT)	**6**	**12**

um dos arquivos de teste, através de um projeto fatorial $2^4 \times 30$. Em seguida, os mesmos valores são utilizados para mensurar a influência no tempo total gasto na etapa de treino, através de um projeto fatorial 2^4.

Os valores mínimos (nível -1) e máximos (nível +1) de cada fator, apresentados na Tabela 4.4, foram definidos de forma que cada valor do nível máximo correspondesse ao dobro do valor do nível mínimo. Conforme dito na Seção 4.1, os experimentos foram realizados com 100% e 25% dos exemplos de treino.

Cada um dos experimentos foi executado 30 vezes. A explicação para este valor vem do *Teorema Central do Limite* [Jain, 1991], que afirma que ao selecionar amostras aleatórias de tamanho n de uma população, a distribuição amostral da média das amostras pode ser aproximada pela distribuição normal de probabilidade à medida que o tamanho da amostra se torna maior ($n \geq 30$), independente da forma da distribuição de frequências da população de onde foram retiradas as amostras. *Dessa forma, a distribuição da média amostral é aproximadamente normal e seus valores de média e desvio padrão estão relacionados com os valores de média e desvio padrão da população.* Isto significa que, ao repetir os experimentos 30 vezes, os valores de F1 médio e desvio padrão dos melhores indivíduos estarão relacionados com os respectivos valores da população.

4.2.1 Impacto na Qualidade das Soluções Geradas

Nesta seção, será avaliado o impacto de cada um dos fatores e interações na qualidade das soluções obtidas, utilizando como variável resposta o F1 médio do melhor indivíduo gerado para cada um dos três arquivos de teste. Para isso, realizou-se um projeto

fatorial com replicação $2^4 \times 30$. Ao replicar os experimentos, foi possível mensurar também os percentuais de variação explicada pelos erros experimentais.

Os resultados são apresentados na Tabela 4.6 da seguinte forma: a primeira linha da tabela contém a identificação dos experimentos, a segunda apresenta a configuração do repositório de dados, com os valores de cada um dos fatores (características dos repositórios), a terceira e a quarta linhas apresentam os valores de F1 médio e desvio padrão do melhor indivíduo encontrado em cada um dos arquivos de teste (A, B e C), utilizando 100% e 25% dos exemplos de treino, respectivamente, enquanto a quinta linha apresenta os percentuais de redução e redução média do F1 médio do melhor indivíduo em cada arquivo de teste – valores negativos de percentuais indicam que, ao reduzir a quantidade de exemplos utilizados no treino, houve um aumento no F1 médio. Os valores de F1 obtidos em cada execução do processo evolucionário são apresentados nas Tabelas A.2 a A.7 do Apêndice A.

4.2.1.1 Experimentos Realizados com 100% dos Exemplos de Treino

Ao realizar o projeto fatorial, verificou-se que os fatores que exerceram maior impacto na qualidade das soluções geradas (medida pelo F1 médio do melhor indivíduo em cada uma das 30 execuções do processo de deduplicação) foram o *MRT (Número Máximo de Modificações Realizadas em uma Réplica)* e o *DBO (Número Máximo de Réplicas Criadas a partir de um Registro Original)*. A média do percentual da variação explicada pelo primeiro fator foi de 8,261% (nos três arquivos de avaliação), enquanto a do segundo fator foi de 7,191%. A interação entre os dois fatores também foi responsável por uma variação média considerável na qualidade dos indivíduos: 7,277%. Os efeitos dos dois fatores e da interação entre eles são significativos com valores de confiança superiores a 87%.

Os demais fatores e interações exerceram impactos menos expressivos, principalmente a interação $DPO \cap DBO \cap MRT$, cujos percentuais de variação foram inferiores a 2% nos três arquivos de teste. Como pode ser visto nas Tabelas A.8 e A.10 do Apêndice A, esta interação não exerceu influência alguma nos resultados obtidos no arquivo de teste A, no que diz respeito à qualidade dos indivíduos gerados. Essas mesmas tabelas apresentam os demais valores de percentuais de variação explicada por cada fator e interação, além dos intervalos e valores máximos de confiança para os experimentos do projeto fatorial que utilizam 100% dos exemplos de treino. Os valores máximos de confiança foram definidos de forma que nenhum dos intervalos de confiança incluísse o valor zero. Isso implica que os efeitos de cada fator são significativos com um determinado nível de confiança [Jain, 1991].

Os percentuais totais explicados pelos fatores e interações foram de 67,066%, 61,513% e 58,068%, nos arquivos de teste A, B e C, respectivamente, sendo os percentuais restantes atribuídos a erros experimentais. A existência de parâmetros não controlados pode ser uma das razões para a observância de erros experimentais, uma vez que o projeto fatorial avalia apenas o impacto dos fatores e interações definidos pelo projetista.

Além disso, pode ser que a quantidade de execuções definida para experimentação não tenha sido suficiente para a obtenção de soluções mais estáveis, ou seja, os valores de desvio padrão dos melhores indivíduos poderiam ser ainda menores caso o processo evolucionário fosse executado mais vezes. É importante lembrar que, quanto maior for o número de execuções do processo evolucionário, mais estáveis serão os resultados obtidos (F1 médio e desvio padrão do melhor indivíduo). Entretanto, será exigido um tempo superior para a geração das funções de deduplicação. Este problema é reduzido consideravelmente quando se utiliza a abordagem proposta para a seleção automática de exemplos de treino apresentada no Capítulo 3, que utiliza conjuntos de treino de tamanhos reduzidos para a geração dessas funções de forma mais eficiente.

Por fim, sabemos que a programação genética é uma técnica não determinística que pode ser vista como uma heurística adaptativa, o que pode contribuir de alguma forma para a existência de erros experimentais. Mesmo assim, foi possível identificar os fatores (características dos repositórios de dados) que tiveram maior impacto na qualidade das soluções obtidas ao final do processo de geração das funções de deduplicação. A Tabela 4.7 e a Figura 4.5 apresentam esses resultados em detalhes.

4.2.1.2 Experimentos Realizados com 25% dos Exemplos de Treino

No conjunto de experimentos que utilizam apenas um quarto dos exemplos de treino, os resultados foram bastante diferentes dos obtidos nos experimentos descritos na seção anterior. Desta vez, o fator que teve maior impacto na qualidade das soluções foi o *DPO (Proporção de Registros Duplicados por Registro Original)*, com um percentual médio de variação de 7,176%.

Três interações envolvendo esse fator exerceram um impacto ainda maior nos resultados obtidos: $DPO \cap DBO \cap MRT$, $DPO \cap DBO \cap MUA$ e $DPO \cap MRT$, com percentuais médios de variação de 10,291%, 7,981% e 7.863%, respectivamente. Os efeitos desse fator e das respectivas interações são significativos com elevados níveis de confiança, sempre superiores a 97%. As Tabelas A.9 e A.11 do Apêndice A apresentam os demais valores de percentuais de variação explicada por cada fator e interação, além dos intervalos e valores máximos de confiança, para os experimentos do projeto fatorial

que utilizam 25% dos exemplos de treino.

Os percentuais totais de variação explicada pelos fatores e interações foram superiores aos do conjunto anterior de experimentos: 72,667%, 71,816% e 68,816%, nos arquivos de teste A, B e C, respectivamente. Os percentuais restantes são atribuídos a erros experimentais. A seleção de conjuntos reduzidos de dados para treino, além de tornar a execução desta etapa mais eficiente – mas ainda assim obtendo bons resultados – é responsável por uma redução no percentual da variação não explicada pelos fatores e interações devido a erros experimentais. Esses percentuais podem ser visualizados na Figura 4.5.

4.2.1.3 Conclusões sobre o Impacto na Qualidade das Soluções Geradas

Os experimentos do projeto fatorial que utilizaram 100% dos exemplos de treino tiveram resultados bastante diferentes dos obtidos nos experimentos que utilizam apenas 25% dos exemplos para a etapa de treino.

No primeiro conjunto de experimentos, os fatores que tiveram maior impacto foram o *MRT (Número Máximo de Modificações Realizadas em uma Réplica)* e o *DBO (Número Máximo de Réplicas Criadas a partir de um Registro Original)*. Dessa forma, espera-se que, quanto maior for o nível de "sujeira" dentro dos registros do repositório de dados, mais difícil será a tarefa de identificação de réplicas utilizando a abordagem baseada em PG. Por outro lado, o fator *MUA (Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica)* não influenciou tanto na qualidade das soluções geradas, ao contrário do que ocorreu no projeto fatorial para a avaliação do impacto no tempo de treino, onde o fator foi responsável por 25,2% da variação nos resultados, ficando atrás apenas do fator *DPO (Proporção de Registros Duplicados por Registro Original)*.

Entretanto, o fator que teve maior impacto no segundo conjunto de experimentos foi o *DPO*, enquanto o *MRT*, fator que teve maior impacto nos resultados do primeiro conjunto de experimentos, foi o que menos influenciou a qualidade das soluções no segundo conjunto de experimentos. Diversas interações envolvendo o fator *DPO* também exerceram um grande impacto nos resultados obtidos, evidenciando ainda mais a importância deste fator no conjunto de experimentos. Dessa forma, espera-se que, quanto maior for a quantidade de réplicas no repositório de dados, mais complicado será o processo de identificação de réplicas utilizando PG. As tabelas referentes aos experimentos do projeto fatorial podem ser encontradas no Apêndice A.

Tabela 4.5. Tempo Gasto na Etapa de Treino do Processo de Deduplicação de Registros, em cada Experimento do Projeto Fatorial 2^4, com 100% e 25% dos Exemplos de Treino.

Experimento	1	2	3	4	5	6	7	8
Configuração								
DPO	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25
DBO	2	2	2	2	4	4	4	4
MUA	3	3	6	6	3	3	6	6
MRT	6	12	6	12	6	12	6	12
Tempo de Treino (hs) - 100%	**71,2**	**64,3**	**67,2**	**64,9**	**62,2**	**69,2**	**72,2**	**69,5**
Tempo de Treino (hs) - 25%	**20,1**	**17,6**	**16,2**	**14,7**	**16,5**	**18,0**	**15,6**	**19,4**
Redução Percentual	71,8%	72,6%	75,9%	77,3%	73,5%	74,0%	78,4%	72,1%

Experimento	9	10	11	12	13	14	15	16
Configuração								
DPO	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
DBO	2	2	2	2	4	4	4	4
MUA	3	3	6	6	3	3	6	6
MRT	6	12	6	12	6	12	6	12
Tempo de Treino (hs) - 100%	**70,7**	**66,6**	**76,7**	**82,4**	**63,3**	**74,2**	**76,0**	**77,3**
Tempo de Treino (hs) - 25%	**16,8**	**19,7**	**15,5**	**19,2**	**17,6**	**19,1**	**18,2**	**20,7**
Redução Percentual	76,2%	70,4%	79,8%	76,7%	72,2%	74,3%	76,1%	73,2%

4.2.2 Impacto no Tempo de Treino

Para avaliar o impacto dos fatores e interações no tempo gasto na etapa de treino do processo de deduplicação de registros utilizando PG, realizou-se um projeto fatorial 2^4 sem replicação. Dessa forma, foram realizados 16 experimentos, um para cada combinação dos quatro fatores. A nomenclatura utilizada para descrever as configurações dos experimentos obedece às siglas dispostas na primeira coluna da Tabela 4.4.

A Tabela 4.5, por sua vez, apresenta o tempo gasto – em horas – na etapa de treino de cada um dos experimentos realizados no projeto fatorial 2^4, quando se utiliza 100% e 25% dos exemplos para treino. Ao analisar os resultados, fica evidente que a seleção de um conjunto reduzido de exemplos de treino agiliza o processo de deduplicação de registros. Em todos os experimentos realizados, a redução no tempo de treino foi superior a 70%.

4.2.2.1 Experimentos Realizados com 100% dos Exemplos de Treino

De posse dos valores de tempo de treino dos experimentos realizados, foi possível calcular o percentual de influência de cada um dos fatores e interações nos resultados. A importância de um fator é medida pela proporção da variação total na resposta explicada por ele [Jain, 1991].

O *DPO (Proporção de Registros Duplicados por Registro Original)* foi o fator que exerceu maior impacto nos resultados, sendo responsável por 27,5% da variação do tempo necessário para treino, seguido pelo fator *MUA (Número Máximo de*

Modificações Realizadas em cada Atributo de uma Réplica), responsável por 25,2% da variação do tempo de treino. Já a interação entre esses dois fatores foi responsável por 12,0% da variação. É bastante evidente o quanto esses dois fatores influenciam o tempo gasto para treino (64,7%).

Por outro lado, o *MRT (Número Máximo de Modificações Realizadas em uma Réplica)* explica apenas 1,0% da variação, enquanto o *DBO (Número Máximo de Réplicas Criadas a partir de um Registro Original)* praticamente não influencia nos resultados. Mais informações sobre os demais percentuais de influência de cada fator e interação do projeto fatorial podem ser obtidas na Tabela A.1 do Apêndice A.

4.2.2.2 Experimentos Realizados com 25% dos Exemplos de Treino

Nos experimentos que utilizam um conjunto reduzido de exemplos de treino (25%), verificou-se um resultado diferente. O fator que teve maior impacto nos tempos de treino foi o *MRT*, responsável por 18,2% da variação no tempo de treino, seguido pelo *DPO (Proporção de Registros Duplicados por Registro Original)*, o fator que mais impacto teve nos resultados do conjunto de experimentos anterior, agora responsável por 9,7% da variação.

As duas interações que mais influenciaram nos tempos de treino também envolveram estes dois fatores: as interações $DPO \cap DBO \cap MRT$ e $DPO \cap MRT$ responderam por 18,2% e 11,1% da variação, respectivamente. Logo, a soma das participações dos dois fatores e das duas interações chegou a 57,2% da variação nos tempos de treino, valor consideravelmente elevado, mostrando a importância destes dois fatores neste conjunto de experimentos.

4.2.2.3 Conclusões sobre o Impacto no Tempo de Treino

Aparentemente, o fator que mais afeta o tempo de treino do processo de deduplicação de registros baseado em PG é o *DPO*, uma vez que apareceu como um dos mais influentes nos dois conjuntos de experimentos, realizados com diferentes percentuais de exemplos de treino.

Já o fator *DBO* teve influência praticamente nula nos resultados obtidos, o que leva a concluir que pouco importa a quantidade de réplicas existentes para um registro original. Assim, essa característica do repositório de dados não influencia de forma significativa o tempo de treino do processo de deduplicação de registros utilizando PG. A Tabela A.1 do Apêndice A apresenta os demais resultados referentes à experimentação realizada nesta seção.

4.2.3 Conclusões Finais do Projeto Fatorial

A execução de um projeto fatorial utilizando duas variáveis resposta (F1 médio do melhor indivíduo e tempo de treino) e dois conjuntos de treino de diferentes tamanhos (utilizando 100% e 25% dos exemplos de treino) possibilitou a realização de um importante estudo sobre as características dos repositórios de dados que possam tornar a tarefa de deduplicação de registros utilizando PG mais ou menos custosa. Algumas conclusões importantes são apresentadas nesta seção.

Para os experimentos que utilizaram 100% dos exemplos de treino, o projeto fatorial demonstrou que os fatores *DPO (Proporção de Registros Duplicados por Registro Original)* e *MUA (Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica)* são os que exercem maior impacto no tempo de treino do processo de deduplicação de registros utilizando PG. Ao avaliar o impacto dos fatores na qualidade dos indivíduos, o resultado foi completamente diferente. Os fatores *DBO (Número Máximo de Réplicas Criadas a partir de um Registro Original)* e *MRT (Número Máximo de Modificações Realizadas em uma Réplica)*, que apresentaram percentuais de variação praticamente nulos no projeto fatorial realizado para avaliar o impacto no tempo de treino, são responsáveis pelos maiores percentuais de variação na qualidade das soluções obtidas ao final do processo de geração das funções de deduplicação.

Ao utilizar apenas 25% dos exemplos de treino, o projeto fatorial apresentou resultados diferentes. Verificou-se que os fatores *DPO* e *MRT* são os que exercem maior impacto no tempo para treino, juntamente com algumas interações das quais os dois fazem parte. Já na avaliação da qualidade das soluções geradas, o fator mais relevante foi o DPO, enquanto o MRT pouco influenciou nos resultados.

Dessa forma, constatou-se que a importância de cada fator e interação na qualidade das soluções geradas varia de acordo com o tamanho do conjunto de exemplos utilizado na etapa de treino do processo de deduplicação de registros utilizando PG. Ao utilizar conjuntos de treino maiores, a principal dificuldade para a deduplicação se encontra no nível geral de "sujeira" dos registros que compõem o repositório de dados, ao passo que, utilizando conjuntos de treino reduzidos, a dificuldade passa a se concentrar no nível de "sujeira" do próprio repositório, no que diz respeito à proporção de registros duplicados por registro original.

Além disso, ficou evidente que o *DPO* é o fator que mais afeta o tempo de treino do processo de deduplicação de registros utilizando PG, ou seja, quanto maior for a proporção de réplicas por registro original, maior será a quantidade de réplicas no repositório de dados, acarretando um aumento considerável no tempo necessário para a execução da etapa de treino do processo de deduplicação de registros utilizando PG.

Tabela 4.6. F1 Médio e Desvio Padrão do Melhor Indivíduo em cada Arquivo de Teste (A, B e C), para os Experimentos que utilizam 100% e 25% dos Exemplos de Treino.

Experimento	1	2	3	4
Configuração				
DPO	0,25	0,25	0,25	0,25
DBO	2	2	2	2
MUA	3	3	6	6
MRT	6	12	6	12
100% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**923/941/910**	**870/883/860**	**897/932/903**	**855/875/873**
Desvio Padrão ($\times 10^{-3}$)	**064/055/111**	**096/078/102**	**112/070/092**	**123/106/090**
25% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**919/933/924**	**844/865/843**	**888/917/877**	**805/820/847**
Desvio Padrão ($\times 10^{-3}$)	**066/056/096**	**092/066/105**	**080/057/071**	**109/101/079**
Redução do F1	0,4%/0,8%/-1,5%	3,0%/2,0%/2,0%	1,0%/1,6%/2,9%	5,8%/6,3%/3,0%
Redução Média do F1	-0,1%	2,3%	1,8%	4,0%

Experimento	5	6	7	8
Configuração				
DPO	0,25	0,25	0,25	0,25
DBO	4	4	4	4
MUA	3	3	6	6
MRT	6	12	6	12
100% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**906/905/913**	**861/860/879**	**933/920/929**	**875/894/904**
Desvio Padrão ($\times 10^{-3}$)	**063/072/062**	**124/127/111**	**068/070/055**	**075/065/062**
25% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**915/922/919**	**836/841/857**	**900/886/901**	**859/878/898**
Desvio Padrão ($\times 10^{-3}$)	**064/065/058**	**134/135/119**	**073/070/053**	**090/086/087**
Redução do F1	-1,0%/-1,9%/-0,7%	2,9%/2,2%/2,5%	3,5%/3,7%/3,0%	1,8%/1,8%/0,7%
Redução Média do F1	-1,2%	2,5%	3,4%	1,4%

Experimento	9	10	11	12
Configuração				
DPO	0,50	0,50	0,50	0,50
DBO	2	2	2	2
MUA	3	3	6	6
MRT	6	12	6	12
100% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**941/919/940**	**869/889/891**	**949/955/954**	**924/922/928**
Desvio Padrão ($\times 10^{-3}$)	**061/065/044**	**113/100/079**	**069/058/067**	**063/063/062**
25% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**921/900/927**	**900/912/911**	**884/896/887**	**907/898/909**
Desvio Padrão ($\times 10^{-3}$)	**064/072/047**	**096/085/065**	**085/073/083**	**058/064/061**
Redução do F1	2,1%/2,1%/1,4%	-3,6%/-2,6%/-2,2%	6,8%/6,2%/7,0%	1,8%/2,6%/2,0%
Redução Média do F1	1,9%	-2,8%	6,7%	2,1%

Experimento	13	14	15	16
Configuração				
DPO	0,50	0,50	0,50	0,50
DBO	4	4	4	4
MUA	3	3	6	6
MRT	6	12	6	12
100% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**923/913/914**	**864/896/917**	**937/929/946**	**923/920/911**
Desvio Padrão ($\times 10^{-3}$)	**056/068/053**	**088/083/077**	**064/072/050**	**069/074/085**
25% - A/B/C				
F1 Médio ($\times 10^{-3}$)	**927/913/914**	**844/880/898**	**927/922/934**	**927/925/918**
Desvio Padrão ($\times 10^{-3}$)	**057/072/054**	**087/085/078**	**067/072/048**	**058/063/072**
Redução do F1	-0,4%/0,0%/0,0%	2,3%/1,8%/2,1%	1,1%/0,8%/1,3%	-0,4%/-0,5%/-0,8%
Redução Média do F1	-0,1%	2,1%	1,1%	-0,6%

Tabela 4.7. Valores dos Percentuais de Variação Explicada pelos Fatores e Interações mais Relevantes do Projeto Fatorial $2^4 \times 30$, utilizando 100% e 25% dos Exemplos de Treino.

100% Fatores/Interações	Arquivo de Teste A	Arquivo de Teste B	Arquivo de Teste C	*Percentual Médio*
MRT	13,905%	8,170%	2,708%	*8,261%*
MRT ∩ *DBO*	0,202%	10,434%	11,194%	*7,277%*
DBO	18,602%	2,571%	0,401%	*7,191%*
Outros	34,357%	40,338%	43,765%	*39,487%*
Erros	32,934%	38,487%	41,932%	*37,784%*
Total	100,000%	100,000%	100,000%	*100,000%*

25% Fatores/Interações	Arquivo de Teste A	Arquivo de Teste B	Arquivo de Teste C	*Percentual Médio*
DPO ∩ *DBO* ∩ *MRT*	4,939%	15,847%	10,086%	*10,291%*
DPO ∩ *DBO* ∩ *MUA*	22,976%	0,603%	0,364%	*7,981%*
DPO ∩ *MRT*	2,326%	15,590%	5,673%	*7,863%*
DPO	8,271%	9,255%	4,003%	*7,176%*
Outros	34,155%	30,521%	48,690%	*37,789%*
Erros	27,333%	28,184%	31,184%	*28,900%*
Total	100,000%	100,000%	100,000%	*100,000%*

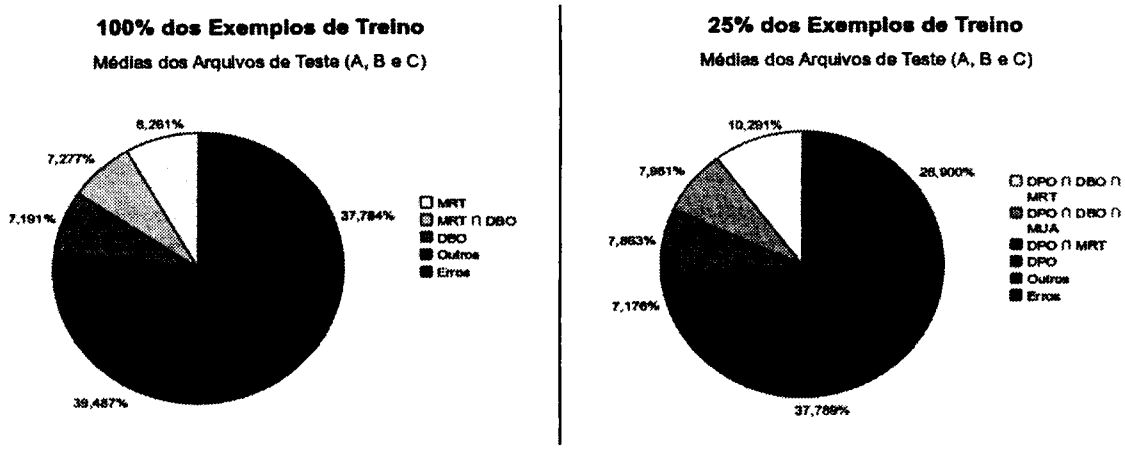


Figura 4.5. Representação Visual das Médias dos Percentuais de Variação Explicada pelos Fatores e Interações mais Relevantes do Projeto Fatorial $2^4 \times 30$, cujos valores são apresentados na Tabela 4.7.

Capítulo 5

Conclusões e Trabalhos Futuros

A identificação e remoção de réplicas em repositórios de dados é de extrema importância para a manutenção da qualidade das informações disponibilizadas pelos serviços atuais de armazenamento de dados. Sistemas que dependem da integridade dos dados para oferecer serviços de alta qualidade, como bibliotecas digitais e sistemas de comércio eletrônico, podem ser afetados pela existência de réplicas ou quase-réplicas em seus repositórios. Logo, tem sido feito um grande esforço no desenvolvimento de métodos efetivos e eficientes para a remoção de réplicas em grandes repositórios de dados [Bell & Dravis, 2006; Geer, 2008].

Por ser uma tarefa complexa, cujo tratamento requer muito tempo e poder de processamento devido à necessidade de se comparar grandes quantidades de registros, técnicas de aprendizagem de máquina têm sido utilizadas com sucesso no tratamento do problema de deduplicação. Entretanto, tais técnicas necessitam de uma etapa de treino na qual exemplos gerados manualmente são usados para aprendizado dos padrões de duplicação. Esta geração manual dificulta a utilização dessas técnicas em muitas situações reais devido ao custo e tempo necessários para se criar o conjunto de exemplos de treino. Nesta dissertação, é proposta uma abordagem baseada em uma técnica determinística para sugerir, de forma automática, exemplos de treino para um método de deduplicação de registros baseado em programação genética, viabilizando a utilização dessa técnica em aplicações reais e de grande porte.

Os resultados do estudo experimental apresentado na Seção 3.3 mostram que é possível utilizar uma quantidade reduzida de exemplos de treino sem que a qualidade dos indivíduos (funções de deduplicação) gerados ao final do processo de deduplicação seja consideravelmente afetada. Para o repositório de dados utilizado na experimentação descrita naquela seção da dissertação, a utilização de 10% dos pares de registros positivos e negativos já foi suficiente para a obtenção de resultados satisfatórios,

reduzindo significativamente o tempo necessário para treino. Além disso, um método determinístico foi utilizado com sucesso para a seleção automática de exemplos de treino para o processo de deduplicação de registros utilizando PG, auxiliando na validação da abordagem de seleção de exemplos proposta neste trabalho.

No Capítulo 4, experimentos iniciais mostraram que a utilização de 25% dos exemplos de treino é suficiente para a deduplicação de registros em repositórios de dados com um grau de dificuldade (de deduplicação) mais elevado. Em seguida, foi realizado um projeto fatorial – utilizando 100% e 25% dos exemplos de treino – para identificar as características dos repositórios de dados que facilitam e dificultam a utilização da abordagem proposta para a seleção dos exemplos de treino, mostrando quais fatores exercem maior impacto nos resultados obtidos, com relação à qualidade dos indivíduos gerados e ao tempo gasto na etapa de treino do método de deduplicação de registros utilizando PG.

Os resultados reforçaram a idéia de que a utilização de um conjunto reduzido de exemplos de treino permite a obtenção de resultados satisfatórios, com a vantagem de se demandar um tempo de treino consideravelmente inferior, o que é essencial em termos de escalabilidade. Em todos os experimentos que utilizaram 25% dos exemplos de treino, os tempos de treino tiveram uma redução superior a 70%.

O projeto fatorial realizado para mensurar o impacto dos fatores no tempo de treino mostrou que o *DPO (Proporção de Registros Duplicados por Registro Original)* foi o fator que mais afetou esta variável resposta, aparecendo entre os dois fatores/interações mais influentes nos dois conjuntos de experimentos (utilizando 25% e 100% dos exemplos de treino), enquanto o fator *DBO (Número Máximo de Réplicas Criadas a partir de um Registro Original)* praticamente não teve influência nos resultados obtidos.

A avaliação do impacto na qualidade das soluções geradas apresentou resultados diferentes. Para os experimentos realizados com 100% dos exemplos de treino, os fatores que mais afetaram a qualidade dos indivíduos gerados foram o *MRT (Número Máximo de Modificações Realizadas em uma Réplica)* e o *DBO*. Já nos experimentos que utilizaram um conjunto de treino reduzido (25%), o fator que mais afetou os resultados foi o *DPO*, enquanto o *MRT* foi um dos que menos influenciou. Logo, verificou-se que a importância de cada fator e interação na qualidade dos indivíduos gerados varia de acordo com o tamanho do conjunto de treino utilizado no processo de deduplicação de registros baseado em PG.

Além disso, ao utilizar um conjunto de treino reduzido, os fatores e interações foram capazes de explicar um percentual maior da variação na qualidade das soluções obtidas, de forma que os erros experimentais foram responsáveis por percentuais médios

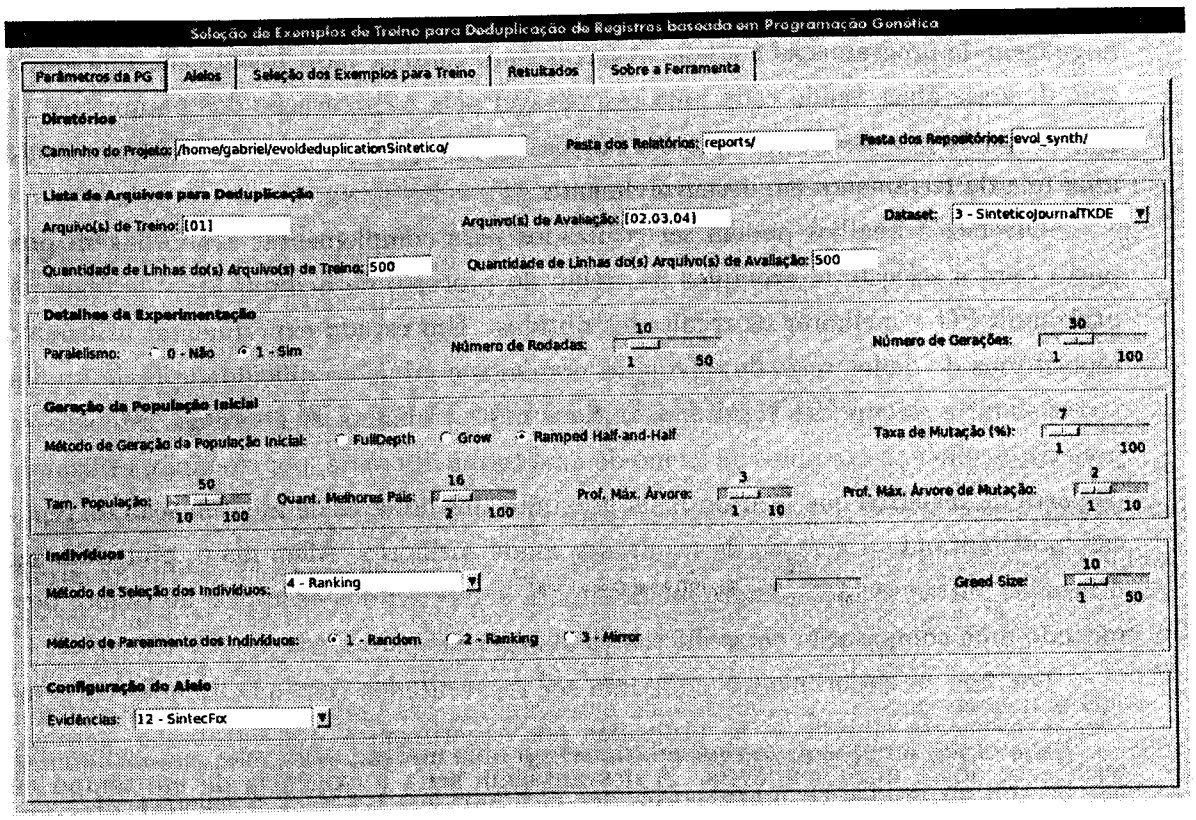


Figura 5.1. Tela da Ferramenta em Desenvolvimento – Aba Referente à Configuração dos Parâmetros da PG.

(de variação) ainda mais reduzidos. Este fato mostra novamente que a utilização de um conjunto de treino maior não implica que o processo de deduplicação obterá resultados melhores. Pelo contrário. A partir de certo ponto, a inclusão de registros no conjunto de treino pouco contribui para a melhoria da qualidade das soluções obtidas, gerando um aumento considerável no tempo de treino e tornando o processo de geração das funções de deduplicação mais custoso. Dessa forma, é importante utilizar apenas a quantidade realmente necessária de exemplos para a etapa de treino, uma vez que, conforme dito anteriormente, a geração dos indivíduos (funções de deduplicação) é uma tarefa que exige muito tempo e esforço computacional, devendo ser realizada da forma mais eficiente possível.

Atualmente, uma ferramenta com interface gráfica (*GUI*) vem sendo desenvolvida para facilitar o processo de geração das funções de deduplicação do método baseado em PG. No momento, o usuário precisa alterar diversos arquivos escritos em Python para configurar o processo de geração dessas funções, exigindo um bom conhecimento da sintaxe dessa linguagem e restringindo a utilização da ferramenta a usuários mais experientes. Uma interface gráfica facilitará o uso da abordagem proposta por usuários

que não estejam tão habituados com o ambiente de deduplicação de registros e com a linguagem de programação Python. A ferramenta consiste basicamente em uma janela com diversas abas, sendo cada uma responsável pela configuração dos parâmetros de uma etapa do processo de geração das funções de deduplicação. A Figura 5.1 apresenta uma tela da ferramenta em desenvolvimento.

Diversos trabalhos podem ser realizados para complementar a abordagem proposta para a seleção de exemplos de treino do processo de deduplicação de registros utilizando PG e melhorar os resultados obtidos. Um estudo experimental utilizando repositórios de dados reais de diferentes domínios e graus de dificuldade pode ajudar a consolidar e estender os resultados obtidos neste trabalho. A abordagem de seleção poderia escolher os exemplos de treino de uma forma diferente, por exemplo, utilizando os pares de registros que ficaram mais próximos dos limiares de identificação de réplicas, possibilitando a geração de resultados ainda melhores. Além disso, poderiam ser utilizadas outras técnicas determinísticas para a geração dos exemplos de treino, permitindo uma comparação entre diferentes abordagens para a seleção dos exemplos.

Por fim, os experimentos poderiam ser reexecutados com quantidades de gerações e execuções superiores às utilizadas nos experimentos deste trabalho, gerando resultados ainda mais confiáveis. A desvantagem seria a exigência de um tempo de experimentação maior, visto que o número de gerações é o parâmetro de controle que tem maior impacto no tempo de treino do processo de deduplicação de registros utilizando PG [de Carvalho et al., 2008b].

Além disso, existe um outro problema com relação ao número de gerações definido para experimentação. Conforme dito anteriormente, quanto maior for o valor deste parâmetro, mais tempo será necessário para a realização da etapa de treino. Se o experimento levar mais tempo do que o necessário para ser executado, as soluções podem se tornar muito especializadas naquele conjunto de registros destinado para treino. Por outro lado, se for definido um valor muito reduzido, pode ser que o processo de geração das funções de deduplicação não consiga encontrar boas soluções. Este *trade-off* evidencia ainda mais a importância de se fazer boas escolhas de parâmetros de controle, com o objetivo de se encontrar boas soluções no menor tempo possível [de Carvalho et al., 2008b].

Apêndice A

Tabelas do Projeto Fatorial

Neste apêndice, são apresentadas as demais tabelas referentes aos experimentos do projeto fatorial descrito no Capítulo 4. Primeiramente, os percentuais de variação explicada pelos fatores e interações do projeto fatorial realizado para a avaliação do impacto no tempo de treino são apresentados na Tabela A.1. Apenas os efeitos considerados mais relevantes haviam sido apresentados anteriormente na Seção 4.2.2.

As Tabelas A.2, A.3 e A.4, por sua vez, apresentam os valores de F1, F1 médio e desvio padrão do melhor indivíduo encontrado nos arquivos de teste A, B e C, respectivamente, para os experimentos que utilizam 100% dos exemplos de treino. As Tabelas A.5, A.6 e A.7 apresentam os respectivos valores para os experimentos que utilizam um conjunto de exemplos de treino reduzido, 25%. Por questões de espaço físico, as tabelas apresentam apenas o número de identificação dos experimentos (primeira coluna da tabela), sem a descrição da configuração de cada um deles. As configurações podem ser vistas nas Tabelas 4.5 ou 4.6 do Capítulo 4. A segunda coluna apresenta o valor de F1 obtido em cada uma das trinta execuções do processo evolucionário, enquanto a terceira e quarta colunas exibem os valores de F1 médio e desvio padrão obtidos em cada um dos experimentos, respectivamente.

Já as Tabelas A.8 e A.9 apresentam os percentuais de variação explicada pelos fatores e interações do projeto fatorial para avaliação do impacto na qualidade das soluções geradas, para os experimentos que utilizam 100% e 25% dos exemplos de treino, respectivamente. Os principais resultados foram apresentados na Seção 4.2.1.

Por fim, as Tabelas A.10 e A.11 apresentam os valores de intervalo e nível máximo de confiança para os efeitos de cada fator e interação do projeto fatorial para avaliação do impacto na qualidade das soluções geradas, utilizando 100% e 25% dos exemplos de treino, respectivamente. Estes valores informam o nível máximo de confiança com que é possível afirmar que o coeficiente de um fator ou interação é significativo.

Tabela A.1. Avaliação do Impacto no Tempo de Treino – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **100% e 25% dos Exemplos de Treino.**

Fator/Interação	100% dos Exemplos de Treino	25% dos Exemplos de Treino
DPO	*27,5%*	*9,7%*
DBO	0,0%	3,6%
MUA	*25,2%*	4,5%
MRT	1,0%	*18,2%*
$DPO \cap DBO$	1,6%	1,6%
$DPO \cap MUA$	*12,0%*	5,8%
$DPO \cap MRT$	4,4%	*11,1%*
$DBO \cap MUA$	0,8%	16,4%
$DBO \cap MRT$	7,4%	5,8%
$MUA \cap MRT$	0,3%	3,2%
$DPO \cap DBO \cap MUA$	4,9%	1,4%
$DPO \cap DBO \cap MRT$	0,1%	18,2%
$DBO \cap MUA \cap MRT$	14,5%	0,3%
$DPO \cap DBO \cap MUA \cap MRT$	0,3%	0,2%
Total	100%	100%

Legenda
DPO: Proporção de Registros Duplicados por Registro Original
DBO: Número Máximo de Réplicas Criadas a partir de um Registro Original
MUA: Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica
MRT: Número Máximo de Modificações Realizadas em uma Réplica

Tabela A.2. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste A**, utilizando **100% dos Exemplos de Treino**.

100% dos Exemplos de Treino Arquivo de Teste A			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	963,973,969,835,973,977,973,835,958,977,966,835,835,835,941, 835,982,835,835,950,973,977,977,981,835,973,835,923,969,957	0,923	0,064
2	767,916,963,964,872,930,729,946,952,949,825,935,913,751,936, 887,956,966,873,729,939,729,970,938,729,729,904,729,951,729	0,870	0,096
3	961,988,730,730,988,941,730,730,730,928,964,992,730,926,980, 964,961,961,730,976,964,981,959,965,972,730,730,980,996,945	0,897	0,112
4	877,661,810,942,955,915,661,974,661,971,943,927,937,967,928, 661,971,916,955,938,974,925,661,947,661,947,661,772,661,661	0,855	0,123
5	840,943,964,840,907,988,974,840,967,944,974,805,840,977,916, 968,840,976,952,840,840,840,955,840,959,840,840,851,962,955	0,906	0,063
6	938,951,903,649,886,966,655,903,824,968,649,649,967,942,928, 931,886,649,845,878,966,649,957,649,938,926,959,936,958,938	0,861	0,124
7	832,969,832,832,991,832,972,832,991,966,978,914,832,994,997, 832,979,976,985,973,955,884,975,832,954,978,882,1000,981,978	0,933	0,068
8	917,758,912,920,758,929,971,945,949,890,909,927,785,924,939, 900,758,753,873,914,927,904,946,952,893,758,758,758,805,910	0,875	0,075
9	991,987,848,848,984,993,993,991,982,966,848,971,848,848,950, 993,973,996,954,989,996,964,985,848,848,907,848,978,911,998	0,941	0,061
10	965,912,718,941,718,718,833,969,728,718,718,955,970,718,957, 973,946,947,718,966,959,969,718,972,878,887,944,718,961,987	0,869	0,113
11	993,951,804,985,989,987,961,959,936,996,989,993,987,889,951, 991,983,984,804,993,998,993,804,987,991,971,991,804,985,804	0,949	0,069
12	969,959,803,952,969,918,957,803,803,969,960,965,812,853,964, 982,964,944,945,936,911,789,978,924,978,980,953,894,978,903	0,924	0,063
13	861,952,861,954,974,949,981,976,965,861,861,952,861,981,971, 861,907,962,861,861,994,961,973,950,922,861,811,986,986,865	0,923	0,056
14	860,947,879,898,948,967,886,846,702,914,702,919,883,890,702, 915,905,702,845,904,854,937,702,905,899,913,913,702,931,948	0,864	0,088
15	970,975,891,813,962,966,813,964,963,956,967,971,965,961,813, 966,813,950,984,972,813,974,813,952,976,991,962,954,974,978	0,937	0,064
16	969,808,957,966,808,927,988,950,808,973,978,981,850,980,808, 972,978,960,960,808,960,932,944,957,941,969,988,946,808,808	0,923	0,069

Tabela A.3. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste B**, utilizando **100% dos Exemplos de Treino**.

100% dos Exemplos de Treino Arquivo de Teste B			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	981,981,981,867,989,989,977,867,959,985,982,867,867,867,943, 867,993,867,867,947,989,985,989,993,867,981,867,963,966,961	0,941	0,055
2	815,912,963,970,930,930,781,943,939,974,884,945,911,696,930, 930,943,958,868,781,926,781,966,908,781,781,861,781,935,781	0,883	0,078
3	989,981,828,828,985,962,828,828,828,974,989,977,828,924,989, 989,973,985,828,985,989,981,944,985,977,828,828,985,989,958	0,932	0,070
4	917,720,817,951,971,910,720,974,720,963,966,938,945,967,963, 720,960,917,959,958,971,966,720,978,720,952,720,766,720,720	0,875	0,106
5	825,932,971,825,920,986,975,825,983,945,974,838,825,965,927, 962,825,980,953,825,825,825,980,825,968,825,825,828,968,968	0,905	0,072
6	938,915,930,631,891,966,690,920,854,958,631,631,978,924,939, 918,866,631,901,890,956,631,949,631,938,937,920,933,955,938	0,860	0,127
7	818,974,818,818,997,818,974,818,961,958,986,892,818,971,977, 818,977,983,953,983,938,842,949,818,944,965,871,991,977,957	0,920	0,070
8	922, 804,954,917,804,928,951,959,971,899,956,930,812,936,944, 917,804,778,876,909,933,906,959,963,922,804,804,804,796,945	0,894	0,065
9	968,966,821,821,968,973,975,976,968,968,821,947,821,821,940, 968,949,971,950,957,969,935,943,821,821,853,821,962,931,975	0,919	0,065
10	965,926,752, 971,752,752,865,971,789,752,752,971,978,752,961, 971,978,960,752,973,967,980,752,950,877,904,966,752,969,984	0,889	0,100
11	980,949,834,976,991,971,987,972,953,987,981,998,993,879,948, 996,985,983,834,998,998,989,834,985,987,973,991,834,991,834	0,955	0,058
12	961,963,799,930,948,927,955,799,799,971,964,960,797,880,969, 962,960,952,943,918,888,808,961,943,954,975,971,899,965,893	0,922	0,063
13	828,935,828,962,977,957,977,974,977,828,828,961,828,953,967, 828,883,973,828,828,983,959,981,960,939,828,830,977,979,835	0,913	0,068
14	944,970,896,922,939,968,938,877,739,962,739,959,925,907,739, 950,929,739,861,919,918,966,739,930,926,948,930,739,961,971	0,896	0,083
15	970,992,876,795,972,970,795,957,929,923,989,977,950,975,795, 956,795,954,976,982,795,989,795,960,985,985,957,930,972,977	0,929	0,072
16	955,796,979,971,796,918,987,925,796,952,984,975,850,968,796, 970,968,957,970,796,958,919,934,948,937,972,989,928,796,796	0,920	0,074

Tabela A.4. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste C**, utilizando **100% dos Exemplos de Treino**.

100% dos Exemplos de Treino Arquivo de Teste C			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	996,993,993,757,985,989,996,757,971,993,993,757,757,757,975, 757,982,757,757,982,985,978,982,993,757,951,757,985,1000,982	0,910	0,111
2	812,908,952,970,880,949,711,939,943,978,788,963,894,706,918, 916,909,941,861,711,958,711,959,911,711,711,843,711,931,711	0,860	0,102
3	985,973,768,768,969,921,768,768,768,966,973,970,768,916,989, 977,955,969,768,970,974,981,909,981,949,768,768,973,970,905	0,903	0,092
4	874,735,850,902,962,833,735,949,735,969,908,931,923,932,930, 735,958,922,954,919,958,949,735,890,735,941,735,841,735,735	0,873	0,090
5	842,976,950,842,890,991,991,842,969,982,988,887,842,988,935, 969,842,991,936,842,842,842,943,842,960,842,842,875,950,966	0,913	0,062
6	949,944,906,688,901,964,688,920,857,963,688,688,978,958,957, 934,875,688,891,890,975,688,978,688,949,945,950,945,966,949	0,879	0,111
7	852,975,852,852,983,852,977,852,968,947,957,882,852,974,989, 852,958,978,963,986,956,883,950,852,951,971,879,992,961,960	0,929	0,055
8	953,805,947,947,805,930,959,971,974,935,961,916,815,941,949, 937,805,834,904,940,918,920,947,969,929,805,805,805,857,937	0,904	0,062
9	970,959,876,876,970,980,993,980,966,956,876,959,876,876,952, 982,955,984,938,968,975,954,977,876,876,890,876,975,927,984	0,940	0,044
10	958,913,791,945,791,791,863,976,771,791,791,920,947,791,961, 957,967,960,791,957,962,964,791,963,869,879,950,791,958,982	0,891	0,079
11	985,980,813,989,994,963,976,981,974,998,985,1000,987,890,962, 991,996,985,813,998,1000,991,813,968,989,987,989,813,991,813	0,954	0,067
12	980,957,800,947,966,936,962,800,800,966,965,962,814,873,963, 962,968,971,966,937,940,792,969,937,959,971,976,908,971,918	0,928	0,062
13	851,951,851,944,951,942,958,951,951,851,851,947,851,980,952, 851,937,955,851,851,986,951,960,913,934,851,830,954,960,831	0,914	0,053
14	960,972,938,963,964,984,939,907,768,968,768,954,940,943,768, 972,956,768,925,951,915,966,768,946,949,962,960,768,982,971	0,917	0,077
15	982,987,876,854,984,972,854,982,957,950,967,974,966,967,854, 953,854,930,983,984,854,985,854,956,981,990,976,954,976,979	0,946	0,050
16	958,766,950,964,766,934,967,933,766,969,979,966,834,977,766, 963,977,958,960,766,951,942,950,947,951,973,982,941,766,766	0,911	0,085

Tabela A.5. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste A**, utilizando **25% dos Exemplos de Treino.**

25% dos Exemplos de Treino Arquivo de Teste A			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	970,959,835,949,953,941,969,966,899,898,835,924,806,966,835, 902,943,981,943,806,910,806,973,954,949,905,966,806,911,977	0,919	0,066
2	835,893,895,822,729,729,943,952,917,935,931,729,729,729,884, 729,896,883,930,729,886,956,967,729,817,729,952,729,928,819	0,844	0,092
3	803,939,964,926,803,984,803,939,960,972,830,966,952,803,827, 976,949,984,730,960,938,803,803,803,803,803,803,901,972,926	0,888	0,080
4	772,772,780,772,710,772,932,925,772,772,772,877,661,941,968, 959,921,917,772,661,661,985,934,661,904,772,661,772,661,710	0,805	0,109
5	721,958,899,944,938,923,973,840,933,907,917,988,953,953,974, 940,893,840,840,934,959,925,862,840,950,939,840,931,840,926	0,915	0,064
6	815,935,945,943,910,945,907,649,925,936,907,502,870,864,948, 866,649,822,649,649,865,951,649,955,948,927,649,649,913,946	0,836	0,134
7	832,972,832,762,832,970,979,932,832,957,832,982,832,975,955, 832,988,819,832,934,930,982,945,832,841,967,985,832,832,985	0,900	0,073
8	915,897,933,906,706,758,845,944,920,585,758,920,758,807,919, 911,923,800,919,932,935,706,872,910,758,892,882,892,937,931	0,859	0,090
9	980,998,848,980,848,966,848,996,848,980,848,848,848,848,897, 998,967,848,848,975,848,961,993,976,982,993,848,848,971,980	0,921	0,064
10	912,967,951,965,892,945,949,718,718,950,978,960,718,946,989, 718,970,913,718,974,965,958,976,961,878,930,938,896,935,718	0,900	0,096
11	980,985,804,973,996,804,978,964,804,804,804,983,804,987,801, 804,987,985,804,804,804,804,804,982,804,987,804,804,817,956	0,884	0,085
12	918,951,675,803,921,842,950,934,962,949,917,959,953,940,717, 926,946,907,849,950,967,803,910,803,825,969,803,966,903,969	0,907	0,058
13	915,995,861,861,952,861,981,976,861,976,968,861,861,986,861, 861,974,861,969,984,861,952,982,861,982,861,973,950,976,990	0,927	0,057
14	849,702,914,881,923,817,702,871,865,812,855,921,702,881,949, 869,849,702,702,948,857,861,924,702,955,903,904,879,909,702	0,844	0,087
15	919,958,953,963,983,959,873,978,813,978,941,975,972,973,813, 956,955,813,813,767,971,883,934,975,976,976,935,963,813,954	0,927	0,067
16	934,945,967,808,935,808,962,952,964,966,907,951,963,974,956, 961,959,883,808,949,808,950,967,978,908,969,945,952,963,808	0,927	0,058

Tabela A.6. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste B**, utilizando **25% dos Exemplos de Treino**.

25% dos Exemplos de Treino Arquivo de Teste B			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	982,981,867,954,946,921,962,982,943,941,867,897,818,985,867, 925,919,981,970,818,933,818,974,967,930,904,981,818,894,989	0,933	0,056
2	865,844,908,786,781,781,943,935,922,962,919,781,781,781,910, 781,874,911,938,781,902,934,974,781,857,781,938,781,914,824	0,865	0,066
3	852,973,957,944,852,981,852,965,973,965,901,966,969,852,877, 977,946,981,828,973,973,852,852,852,852,852,852,935,973,928	0,917	0,057
4	766,766,798,766,714,766,934,962,766,766,766,900,720,971,960, 963,938,962,766,720,720,948,966,720,901,766,720,766,720,714	0,820	0,101
5	822,986,954,957,945,913,965,825,968,895,962,997,946,989,977, 825,874,825,825,962,963,920,868,825,965,952,825,938,825,952	0,922	0,065
6	875,937,924,952,882,939,934,631,910,927,914,573,927,920,948, 875,631,875,631,631,903,924,631,955,962,924,631,631,896,922	0,841	0,135
7	818,931,818,775,818,948,959,917,818,933,818,962,818,965,944, 818,977,772,818,909,922,974,943,818,912,934,968,818,818,954	0,886	0,070
8	923,925,953,915,719,804,843,971,935,618,804,960,804,816,931, 913,912,836,929,966,948,719,869,945,804,889,885,893,954,962	0,878	0,086
9	961,973,821,982,821,957,821,976,821,950,821,821,821,821,925, 980,959,821,821,966,821,954,978,957,966,975,821,821,922,954	0,900	0,072
10	948,971,957,958,926,965,937,752,752,959,984,956,752,969,985, 752,978,931,752,964,969,966,973,958,887,927,927,906,962,752	0,912	0,085
11	985,983,834,978,989,834,967,964,834,834,834,959,834,980,816, 834,976,983,834,834,834,834,834,980,834,987,834,834,840,945	0,896	0,073
12	932,944,627,799,918,848,947,902,959,920,930,966,967,942,737, 927,938,919,817,949,969,799,871,799,795,947,799,955,899,968	0,898	0,064
13	915,992,828,828,961,828,976,976,828,968,981,828,828,979,828, 828,971,828,969,970,828,957,984,828,981,828,976,936,981,982	0,913	0,072
14	850,739,958,947,977,888,739,944,911,825,935,931,739,926,959, 928,828,739,739,967,869,938,959,739,960,908,953,843,933,739	0,880	0,085
15	875,937,918,947,982,951,903,961,795,980,931,961,989,977,795, 938,926,795,795,784,969,887,944,972,959,959,951,949,795,966	0,922	0,072
16	918,948,982,796,956,796,959,961,932,966,922,935,979,967,945, 960,970,888,796,939,796,955,966,982,913,982,955,942,960,796	0,925	0,063

Tabela A.7. Avaliação do Impacto na Qualidade das Soluções Geradas – Valores de F1 e Desvio Padrão do Melhor Indivíduo no **Arquivo de Teste C**, utilizando **25% dos Exemplos de Treino**.

25% dos Exemplos de Treino Arquivo de Teste C			
Experimento	**Valores de F1 ($\times 10^{-3}$) em cada Execução**	**F1 Médio**	**Desvio Padrão**
1	986,993,757,951,962,974,993,986,948,974,757,945,756,989,757, 966,978,985,978,756,950,756,978,989,966,947,996,756,922,993	0,924	0,096
2	819,902,938,806,711,711,969,949,966,958,921,711,711,711,890, 711,917,906,945,711,904,939,985,711,783,711,925,711,925,843	0,843	0,105
3	797,962,929,884,797,962,797,945,949,941,835,936,930,797,831, 969,894,953,768,924,943,797,797,797,797,797,797,900,953,866	0,877	0,071
4	841,841,847,841,738,841,934,949,841,841,841,932,735,931,931, 946,937,945,841,735,735,926,928,735,893,841,735,841,735,738	0,847	0,079
5	784,949,970,963,966,898,969,842,959,907,919,991,954,972,985, 842,940,842,842,926,973,914,945,842,963,915,842,970,842,946	0,919	0,058
6	853,930,942,969,903,972,924,688,930,921,920,568,905,924,952, 845,688,847,688,688,914,941,688,946,986,934,688,688,914,960	0,857	0,119
7	852,941,852,814,852,956,956,929,852,931,852,954,852,971,921, 852,980,801,852,939,922,938,932,852,916,935,957,852,852,972	0,901	0,053
8	980,931,968,943,684,805,850,968,952,750,805,957,805,864,958, 959,925,864,947,971,977,684,922,977,805,924,913,931,962,965	0,898	0,087
9	980,975,876,982,876,949,876,977,876,966,876,876,876,876,925, 984,954,876,876,970,876,977,982,961,966,977,876,876,948,955	0,927	0,047
10	924,948,943,941,910,958,952,791,791,926,967,927,791,951,973, 791,970,904,791,962,964,945,969,962,892,935,922,910,939,791	0,911	0,065
11	993,989,813,982,998,813,967,923,813,813,813,976,813,976,795, 813,980,989,813,813,813,813,813,983,813,985,813,813,799,969	0,887	0,083
12	945,965,726,800,929,873,941,927,957,948,953,964,969,940,751, 935,962,950,851,957,956,800,885,800,887,912,800,955,930,982	0,909	0,061
13	927,972,851,851,947,851,954,951,851,960,956,851,851,962,851, 851,935,851,938,970,851,938,956,851,952,851,969,950,981,967	0,914	0,054
14	916,768,963,933,967,911,768,922,922,855,931,959,768,922,975, 922,890,768,768,974,925,933,941,768,982,946,952,917,952,768	0,898	0,078
15	905,906,926,946,981,955,874,974,854,985,916,972,987,979,854, 958,945,854,854,749,960,879,943,966,970,977,937,958,854,962	0,934	0,048
16	944,939,965,766,952,766,967,947,942,961,905,942,957,962,967, 965,961,865,766,955,766,950,969,965,914,965,942,948,945,766	0,918	0,072

Tabela A.8. Avaliação do Impacto na Qualidade das Soluções Geradas – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **100% dos Exemplos de Treino**.

100% dos Exemplos de Treino				
Fator/Interação	Arquivo de Teste A	Arquivo de Teste B	Arquivo de Teste C	Percentual Médio
DPO	6,106%	1,755%	4,789%	4,217%
DBO	*18,602%*	*2,571%*	*0,401%*	*7,191%*
MUA	3,560%	7,142%	0,356%	3,686%
MRT	*13,905%*	*8,170%*	*2,708%*	*8,261%*
DPO ∩ *DBO*	0,129%	0,999%	0,662%	0,597%
DPO ∩ *MUA*	0,202%	2,722%	16,558%	6,494%
DPO ∩ *MRT*	0,050%	0,394%	0,144%	0,196%
DBO ∩ *MUA*	2,473%	11,670%	0,016%	4,720%
DBO ∩ *MRT*	*0,202%*	*10,434%*	*11,194%*	*7,277%*
MUA ∩ *MRT*	11,658%	1,094%	0,144%	4,299%
DPO ∩ *DBO* ∩ *MUA*	4,459%	0,519%	5,613%	3,530%
DPO ∩ *DBO* ∩ *MRT*	0,000%	1,880%	1,939%	1,273%
DBO ∩ *MUA* ∩ *MRT*	0,050%	6,654%	10,953%	5,886%
DPO ∩ *DBO* ∩ *MUA* ∩ *MRT*	5,670%	5,509%	2,592%	4,590%
Total - Fatores e Interações	67,066%	61,513%	58,068%	62,216%
Total - Erros Experimentais	32,934%	38,487%	41,932%	37,784%
Total	100,000%	100,000%	100,000%	100,000%

Tabela A.9. Avaliação do Impacto na Qualidade das Soluções Geradas – Percentuais de Variação Explicada por cada Fator e suas Interações, utilizando **25% dos Exemplos de Treino**.

25% dos Exemplos de Treino				
Fator/Interação	Arquivo de Teste A	Arquivo de Teste B	Arquivo de Teste C	Percentual Médio
DPO	*8,271%*	*9,255%*	*4,003%*	*7,176%*
DBO	0,685%	4,987%	7,989%	4,554%
MUA	0,002%	7,216%	1,695%	2,971%
MRT	1,709%	2,832%	3,393%	2,645%
DPO ∩ *DBO*	0,068%	1,937%	1,456%	1,154%
DPO ∩ *MUA*	1,094%	2,121%	5,523%	2,913%
DPO ∩ *MRT*	*2,326%*	*15,590%*	*5,673%*	*7,863%*
DBO ∩ *MUA*	0,093%	2,028%	8,169%	3,430%
DBO ∩ *MRT*	4,018%	5,280%	6,617%	5,305%
MUA ∩ *MRT*	8,271%	1,434%	9,885%	6,530%
DPO ∩ *DBO* ∩ *MUA*	*22,976%*	*0,603%*	*0,364%*	*7,981%*
DPO ∩ *DBO* ∩ *MRT*	*4,939%*	*15,847%*	*10,086%*	*10,291%*
DBO ∩ *MUA* ∩ *MRT*	5,134%	2,619%	2,727%	3,493%
DPO ∩ *DBO* ∩ *MUA* ∩ *MRT*	13,081%	0,067%	1,236%	4,795%
Total - Fatores e Interações	72,667%	71,816%	68,816%	71,100%
Total - Erros Experimentais	27,333%	28,184%	31,184%	28,900%
Total	100,000%	100,000%	100,000%	100,000%

Legenda
DPO: Proporção de Registros Duplicados por Registro Original
DBO: Número Máximo de Réplicas Criadas a partir de um Registro Original
MUA: Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica
MRT: Número Máximo de Modificações Realizadas em uma Réplica

Tabela A.10. Avaliação do Impacto na Qualidade das Soluções Geradas – Intervalos de Confiança (Limites Inferior e Superior) e Nível Máximo de Confiança para os Efeitos de cada Fator e Interação do Projeto Fatorial, em cada Arquivo de Teste (A, B e C), utilizando **100% dos Exemplos de Treino**.

100% dos Exemplos de treino						
Fator/Interação	Int. Conf. (Inf : Sup) (A)	Conf. Máx.(%) (A)	Int. Conf. (Inf : Sup) (B)	Conf. Máx.(%) (B)	Int. Conf. (Inf : Sup) (C)	Conf. Máx.(%) (C)
0	(0,9252 : 0,9311)	99,98	(0,9191 : 0,9250)	99,98	(0,9282 : 0,9364)	99,98
DPO	(0,0039 : 0,0098)	99,98	(0,0006 : 0,0065)	99,98	(0,0035 : 0,0117)	99,98
DBO	*(-0,0149 : -0,0091)*	*99,98*	*(-0,0073 : -0,0013)*	*99,98*	*(-0,0042 : -0,0002)*	*92,80*
MUA	(0,0023 : 0,0082)	99,98	(0,0042 : 0,0102)	99,98	(0,0001 : 0,0041)	92,80
MRT	*(-0,0133 : -0,0074)*	*99,98*	*(-0,0107 : -0,0047)*	*99,98*	*(-0,0098 : -0,0016)*	*99,98*
DPO ∩ DBO	(-0,0019 : -0,0001)	78,00	(0,0001 : 0,0053)	99,90	(0,0002 : 0,0055)	98,40
DPO ∩ MUA	(0,0001 : 0,0024)	87,00	(0,0015 : 0,0074)	99,98	(0,0100 : 0,0182)	99,98
DPO ∩ MRT	(0,0001 : 0,0012)	52,00	(-0,0033 : -0,0001)	95,40	(0,0001 : 0,0025)	71,00
DBO ∩ MUA	(-0,0073 : -0,0014)	99,98	(-0,0122 : -0,0062)	99,98	(-0,0008 : -0,0001)	24,00
DBO ∩ MRT	*(0,0001 : 0,0024)*	*87,00*	*(0,0057 : 0,0117)*	*99,98*	*(0,0075 : 0,0157)*	*99,98*
MUA ∩ MRT	(-0,0124 : -0,0066)	99,98	(-0,0054 : -0,0002)	99,90	(0,0001 : 0,0025)	74,00
DPO ∩ DBO ∩ MUA	(0,0029 : 0,0088)	99,98	(-0,0038 : -0,0001)	98,00	(-0,0123 : -0,0041)	99,98
DPO ∩ DBO ∩ MRT	(0,0000 : 0,0000)	00,00	(0,0007 : 0,0067)	99,98	(0,0007 : 0,0089)	99,98
DBO ∩ MUA ∩ MRT	(0,0001 : 0,0012)	52,00	(-0,0099 : -0,0040)	99,98	(-0,0155 : -0,0073)	99,98
DPO ∩ DBO ∩ MUA ∩ MRT	(0,0037 : 0,0096)	99,98	(0,0033 : 0,0093)	99,98	(0,0015 : 0,0097)	99,98

Tabela A.11. Avaliação do Impacto na Qualidade das Soluções Geradas – Intervalos de Confiança (Limites Inferior e Superior) e Nível Máximo de Confiança para os Efeitos de cada Fator e Interação do Projeto Fatorial, em cada Arquivo de Teste (A, B e C), utilizando **25% dos Exemplos de Treino**.

25% dos Exemplos de treino						
Fator/Interação	Int. Conf. (Inf : Sup) (A)	Conf. Máx.(%) (A)	Int. Conf. (Inf : Sup) (B)	Conf. Máx.(%) (B)	Int. Conf. (Inf : Sup) (C)	Conf. Máx.(%) (C)
0	(0,9059 : 0,9114)	99,98	(0,9011 : 0,9084)	99,98	(0,9041 : 0,9119)	99,98
DPO	*(0,0055 : 0,0110)*	*99,98*	*(0,0081 : 0,0154)*	*99,98*	*(0,0040 : 0,0118)*	*99,98*
DBO	(0,0001 : 0,0047)	99,98	(0,0050 : 0,0123)	99,98	(0,0072 : 0,0150)	99,98
MUA	(-0,0002 : -0,0001)	8,00	(-0,0140 : -0,067)	99,98	(-0,0090 : -0,0112)	99,98
MRT	(-0,0065 : -0,0010)	99,98	(-0,0102 : -0,0028)	99,98	(-0,0112 : -0,0033)	99,98
DPO ∩ DBO	(-0,0014 : -0,0001)	64,00	(-0,0090 : -0,0017)	99,98	(-0,0087 : -0,0008)	99,98
DPO ∩ MUA	(-0,0057 : -0,0003)	99,98	(0,0020 : 0,0093)	99,98	(0,0053 : 0,0132)	99,98
DPO ∩ MRT	*(0,0016 : 0,0071)*	*99,98*	*(0,0116 : 0,0189)*	*99,98*	*(0,0055 : 0,0133)*	*99,98*
DBO ∩ MUA	(0,0001 : 0,0017)	72,00	(0,0018 : 0,0092)	99,98	(0,0073 : 0,0152)	99,98
DBO ∩ MRT	(-0,0085 : -0,0030)	99,98	(0,0052 : 0,0125)	99,98	(0,0062 : 0,0140)	99,98
MUA ∩ MRT	(0,0055 : 0,0110)	99,98	(-0,0083 : -0,0010)	99,98	(0,0085 : 0,0163)	99,98
DPO ∩ DBO ∩ MUA	*(0,0110 : 0,0165)*	*99,98*	*(0,0001 : 0,0059)*	*99,70*	*(0,0001 : 0,0047)*	*97,20*
DPO ∩ DBO ∩ MRT	*(-0,0091 : -0,0036)*	*99,98*	*(-0,0190 : -0,0117)*	*99,98*	*(-0,0164 : -0,0086)*	*99,98*
DBO ∩ MUA ∩ MRT	(-0,0092 : -0,0038)	99,98	(0,0026 : 0,0099)	99,98	(-0,0104 : -0,0026)	99,98
DPO ∩ DBO ∩ MUA ∩ MRT	(0,0076 : 0,0131)	99,98	(-0,0019 : -0,0001)	64,00	(0,0005 : 0,0083)	99,98

Legenda
DPO: Proporção de Registros Duplicados por Registro Original
DBO: Número Máximo de Réplicas Criadas a partir de um Registro Original
MUA: Número Máximo de Modificações Realizadas em cada Atributo de uma Réplica
MRT: Número Máximo de Modificações Realizadas em uma Réplica

Referências Bibliográficas

Baeza-Yates, R. A. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Banzhaf, W.; Francone, F. D.; Keller, R. E. & Nordin, P. (1998). *Genetic Programming: An Introduction: On The Automatic Evolution of Computer Programs and Its Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Bell, R. & Dravis, F. (2006). Is your data dirty?: (and does that matter?). Technical report, Accenture Whiter Paper. Disponível em http://www.accenture.com.

Bhattacharya, I. & Getoor, L. (2004). Iterative record linkage for cleaning and integration. In *Proceeding of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 11–18, Paris, France.

Bilenko, M.; Mooney, R.; Cohen, W.; Ravikumar, P. & Fienberg, S. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23.

Bilenko, M. & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 39–48, Washington, DC, USA.

Chaudhuri, S.; Ganjam, K.; Ganti, V. & Motwani, R. (2003). Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pp. 313–324, San Diego, CA, USA.

Christen, P. (2005). Probabilistic data generation for deduplication and data linkage. In Gallagher, M.; Hogan, J. M. & Maire, F., editores, *IDEAL*, volume 3578 of *Lecture Notes in Computer Science*, pp. 109–116. Springer.

Christen, P. (2008). Febrl: a freely available record linkage system with a graphical user interface. In *Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management*, pp. 17–25, Wollongong, NSW, Australia.

Cohen, W. W. (2000). Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems*, 18(3):288–321.

de Carvalho, M. G.; Gonçalves, M. A.; Laender, A. H. F. & da Silva, A. S. (2006). Learning to deduplicate. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 41–50, Chapel Hill, NC, USA.

de Carvalho, M. G.; Laender, A. H. F.; Gonçalves, M. A. & da Silva, A. S. (2008a). Replica identification using genetic programming. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pp. 1801–1806, Fortaleza, CE, Brazil.

de Carvalho, M. G.; Laender, A. H. F.; Gonçalves, M. A. & Porto, T. C. (2008b). The impact of parameter setup on a genetic programming approach to record deduplication. In *Proceedings of the 23rd Brazilian Symposium on Databases*, pp. 91–105, Campinas, SP, Brazil.

Elmagarmid, A. K.; Ipeirotis, P. G. & Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16.

Fellegi, I. P. & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Geer, D. (2008). Reducing the storage burden via data deduplication. *Computer*, 41(12):15–17.

Gonçalves, G. S.; de Carvalho, M. G.; Laender, A. H. F. & Gonçalves, M. A. (2009). Seleção automática de exemplos de treino para um método de deduplicação de registros baseado em programação genética. In *XXIV Simpósio Brasileiro de Banco de Dados*, pp. 76–90, Fortaleza, CE, Brasil.

Gu, L. & Baxter, R. (2006). Decision models for record linkage. *Selected Papers from AusDM*, 3755:146–160.

Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York, NY, USA.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA.

Koudas, N.; Sarawagi, S. & Srivastava, D. (2006). Record linkage: similarity measures and algorithms. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pp. 802–803, Chicago, IL, USA.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection.* The MIT Press, Cambridge, MA.

Newcombe, H. B.; Kennedy, J. M.; Axford, S. & James, A. (1959). Automatic linkage of vital records. *Science*, 130(3381):954–959.

Tejada, S.; Knoblock, C. A. & Minton, S. (2001). Learning object identification rules for information integration. *Inf. Syst.*, 26(8):607–633.

Tejada, S.; Knoblock, C. A. & Minton, S. (2002). Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 350–359, Edmonton, AB, Canada.

Verykios, V. S.; Moustakides, G. V. & Elfeky, M. G. (2003). A bayesian decision model for cost optimal record matching. *The VLDB Journal*, 12(1):28–40.

Wheatley, M. (2004). Operation clean data. Technical report, CIO Asia Magazine. Disponível em http://www.cio-asia.com.

Winkler, W. E. (1999). The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau.