

VIVIANE DAL MOLIN DE SOUZA

**OTIMIZAÇÃO MULTIOBJETIVO APLICADA
EM RACIOCÍNIO BASEADO EM CASOS**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutora em Informática.

CURITIBA

2018

VIVIANE DAL MOLIN DE SOUZA

**OTIMIZAÇÃO MULTIOBJETIVO APLICADA
EM RACIOCÍNIO BASEADO EM CASOS**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutora em Informática.

Área de Concentração: *Ciência da Computação*

Orientador: Prof. Dr. Edson Emílio Scalabrin

CURITIBA

2018

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR

S729o 2018	<p>Souza, Viviane Dal Molin de Otimização multiobjetivo aplicada em raciocínio baseado em casos / Viviane Dal Molin de Souza ; orientador: Edson Emilio Scalabrin. - 2018 120 f. : il. ; 30 cm</p> <p>Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2018 Bibliografia: f. 111-120</p> <p>1. Processamento de dados. 2. Raciocínio baseado em casos. 3. Algoritmo. 4. Algoritmos genéticos. I. Scalabrin, Edson Emilio. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título.</p> <p>CDD 22. ed. – 004</p>
---------------	---

Biblioteca Central
Giovanna Carolina Massaneiro dos Santos – CRB 9/1911



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE SESSÃO PÚBLICA

DEFESA DE TESE DE DOUTORADO Nº 55/2018

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGIa PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - PUCPR

Em sessão pública realizada às 14h00 de 18 de Maio de 2018, na Sala 220 – Escola de Negócios, ocorreu a defesa da tese de doutorado intitulada “Otimização Multiobjetivo Aplicada em Raciocínio baseado em Casos” elaborada pela aluna Viviane Dal Molin de Souza, como requisito parcial para a obtenção do título de **Doutor em Informática**, na área de concentração **Ciência da Computação**, perante a banca examinadora composta pelos seguintes membros:

Prof. Dr. Edson Emilio Scalabrin (orientador) - PUCPR

Prof. Dr. Bráulio Coelho Avila – PUCPR

Prof.ª Dr.ª Deborah Ribeiro Carvalho – PUCPR/PPGTS

Prof. Dr. Osmar Betazzi Dordal -

Prof. Dr. Jones Granatyr – UNIVERSIDADE DO CONTESTADO

Após a apresentação da tese pelo aluno e correspondente arguição, a banca examinadora emitiu o seguinte parecer sobre a tese:

Membro	Parecer
Prof. Dr. Edson Emilio Scalabrin	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof. Dr. Bráulio Coelho Avila	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof.ª Dr.ª Deborah Ribeiro Carvalho	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof. Dr. Osmar Betazzi Dordal	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof. Dr. Jones Granatyr	<input checked="" type="checkbox"/> Aprovada () Reprovada

Portanto, conforme as normas regimentais do PPGIa e da PUCPR, a tese foi considerada:

APROVADO

(aprovação condicionada ao atendimento integral das correções e melhorias recomendadas pela banca examinadora, conforme anexo, dentro do prazo regimental)

REPROVADO

E, para constar, lavrou-se a presente ata que vai assinada por todos os membros da banca examinadora. Curitiba, 18 de Maio de 2018.

Prof. Dr. Edson Emilio Scalabrin

Prof. Dr. Bráulio Coelho Avila

Prof.ª Dr.ª Deborah Ribeiro Carvalho

Prof. Dr. Osmar Betazzi Dordal

Prof. Dr. Jones Granatyr



Dedico esse trabalho à minha filha Alana, aos meus pais Luiz e Soeli que são meu porto seguro. Minha família é o meu bem maior.

Agradecimentos

Primeiro quero agradecer a Deus, pois foi Ele quem me sustentou e conduziu em toda minha vida, principalmente nos momentos mais difíceis.

Um agradecimento especial a minha família, meus pais e filha, que me deram o suporte necessário para vencer esta etapa.

Ao meu orientador, prof. Edson Emílio Scalabrin, por todo o suporte e apoio e principalmente por sempre confiar no trabalho desenvolvido. E principalmente por acreditar em mim, quando nem eu mesma acreditava. Seu apoio e conselhos foram fundamentais para eu não desistir.

A CAPES e a UTFPR, por tornar esse trabalho possível.

A todos os amigos que, de alguma forma, participaram dessa caminhada e contribuíram para que esse trabalho fosse realizado. Destes cito alguns que foram fundamentais para esta etapa: Cleverton Vicentini, Denise Maria Vecino Sato, Denise Pereira, Daumar Luiz Pereira, Elaini Simoni Angelotti, Evandro Alberto Zatti, Giulliana Martins de Souza, Jean Paul Bardal, Josy Fatima, Vilmar Abreu Junior e Viviana Cocco Mariani.

Um agradecimento especial a secretaria do nosso curso Cheila Cristina que sempre tem uma palavra amiga e um chimarrão para nos oferecer em sua sala.

A todos os profissionais (médicos psiquiatras, psicólogos, terapeutas...) que me ajudaram em um dos momentos mais difíceis da minha vida, e também a tantas pessoas que conheci pelo caminho que enfrentam seus monstros diários assim como eu.

Resumo

Esta tese apresenta um modelo computacional para políticas de ações para um domínio bem-definido. Cada política bem-sucedida representa uma nova experiência. Cada experiência aprendida amplia, por meio do seu reuso, a capacidade do aprendiz em resolver novos problemas. Este processo pode ser explorado por meio de mecanismos canônicos de raciocínio baseado em casos; em que os casos são de um domínio bem-definido. Aqui, o domínio de estudo se concentra em torno do modal férreo, mais precisamente, na adaptação de políticas ou planos de ações para rebocar trens de carga em vias férreas realistas. Cada plano de ações P é dado por um conjunto ordenado de ações e quando P for aplicado, o resultado deve ser o deslocamento de um trem de um ponto A até um ponto B de uma via férrea. Todavia, a condução de trens de carga é uma tarefa complexa devido às variações nas formações dos trens, nas condições ambientais e, nos perfis das plantas das ferrovias, bem como o grande número de cálculos necessários para determinar cada ação de um plano P . Tal plano deve resultar em uma condução rápida, econômica e segura. Para fazer face a tal complexidade, assume-se que: (i) tarefa de adaptação é essencial para obter planos de ações viáveis e manter fraca dependência do especialista em condução; (ii) a utilização de um método de adaptação multiobjetivo permite obter planos ações eficientes e eficazes. A avaliação da abordagem foi feita por meio de diferentes experimentos, considerando diferentes métricas: (i) redução do tempo de duração de viagem; (ii) economia de combustível; e (iii) eficiência das tarefas de recuperação e adaptação de planos de ações. Os resultados obtidos mostraram que o uso de algoritmo genético multiobjetivo e algoritmo de evolução diferencial multiobjetivo permite gerar políticas de ações viáveis, em particular, no tocante a redução do consumo de combustível; gerando uma economia média de 42% em LTKB (litro por tonelada bruta transportada) em relação a um maquinista. Tal percentagem é expressiva considerando que o objetivo inicial era de obter uma economia média de 6% em LTKB.

Palavras-Chave: Raciocínio baseado em casos, otimização multiobjetivo, plano de condução de trens interurbanos de carga, algoritmo genético, algoritmo de evolução diferencial.

Abstract

This thesis proposes a computational model based on actions policies for a well-defined domain. For every well-succeeded policy appliance, a new instance, called experience, has been stored in the model. Therefore, each experience learned increases the intelligent system capacity for solving new problems. We explored this process using canonical reasoning mechanisms based on cases, where the cases represent the well-defined domain. For this thesis, we used the railway modal as the studied domain, focusing on the adaptation of the policies (or plans) for towing freight trains on real railroads. Each plan P is an ordered set of actions and the execution of the plan results on moving one train through a railroad from point A to B. However, driving a freight train is a complex task, especially because some aspects such as: the order of the cars in the train, the environmental conditions, the distinct railroad profiles and the large number of calculations needed to determine each action in P. Furthermore, every plan P should drive a train safely, minimizing the travel time and consuming the least amount of fuel. In order to properly behave with this complexity we assume that: (i) the adaptation task is essential to obtain viable plans avoiding a strong dependency of the conductor expertise; (ii) the use of a multi-objective adaptation method enables efficient and effective plan generation. We evaluated the computation model based on different experiments, using distinct metrics: (i) travel time; (ii) fuel consumption; (iii) efficiency of the cases recovery and adaptation tasks. The experiments showed that it is possible to generate viable action policies for freight train conduction, using a multiobjective genetic algorithm and a multiobjective differential evolution algorithm as the adaptation method. In addition, the results presented a considerable reduction on the fuel consumption, with a medium reduction of 42% LTKB (liters per ton transported per kilometer). The 42% reduction is very expressive, considering that the initial goal were set to a 6% LTKB reduction.

Keywords: Case-based reasoning, multiobjective optimization, trains driving plan, genetic algorithm, differential evolution algorithm.

Lista de Figuras

Figura 1. Ciclo de um sistema de raciocínio baseado em casos.....	32
Figura 2. Curva de pareto.....	45
Figura 3. Funcionamento algoritmo de evolução diferencial.....	58
Figura 4. Exemplo de malha férrea representada por um grafo rotulado. Cada aresta representa um conjunto de pontos de medida.....	76
Figura 5. Exemplo de malha férrea representada por um grafo rotulado com caminhos entre as estações.....	77
Figura 6. Linha férrea ST , dividida em trechos stn	79
Figura 7. Ciclo clássico CBR destacando contribuições do trabalho.....	81
Figura 8. Representação de um indivíduo.....	83
Figura 9. Exemplo de mutação para a abordagem algoritmo genético.....	84
Figura 10. Exemplo de mutação para a abordagem evolução diferencial.....	84
Figura 11. Cruzamento de 1-ponto.....	85
Figura 12. Processo de adaptação.....	86
Figura 13. Sistema para a geração de planos de ações estruturado em camadas: SE (sistema baseado em heurísticas), AM (aprendizagem de máquina), DCOP (otimização distribuída de restrições), DE (Evolução Diferencial) e GA (Genética).	110

Lista de Tabelas

Tabela 1. Exemplo de descrição de sub-trecho de via férrea com sete pontos de medidas, extraídos do projeto topográfico de uma via férrea.....	78
Tabela 2. Representação da base de casos.....	80
Tabela 3. Configuração de trens dadas pela quantidade de locomotivas, quantidade de vagões e peso.....	91
Tabela 4. Desempenho obtido em termos de LTKB no Experimento 1: usando a abordagem <i>genética multiobjetivo</i>	94
Tabela 5. Desempenho obtido em termos tempo de duração de viagem completa no Experimento 1: usando a abordagem <i>genética multiobjetivo</i>	95
Tabela 6. Desempenho obtido em termos de tempo de duração de viagem completa e em termos de LTKB nos Experimentos 3 e 4: usando a abordagem <i>genética multiobjetivo</i>	95
Tabela 7. Desempenho obtido em termos de taxa de acerto das tarefas de recuperação e de adaptação dos Experimentos 3 e 4: usando a abordagem <i>genética multiobjetivo</i>	98
Tabela 8. Desempenho obtido em termos de LTKB no Experimento 1: usando a abordagem <i>diferencial evolutiva multiobjetivo</i>	99
Tabela 9. Desempenho obtido em termos de tempo de duração de viagem completa no Experimento 1: usando a abordagem <i>diferencial evolutiva multiobjetivo</i>	99
Tabela 10. Desempenho obtido em termos de tempo de duração de viagem completa e em termos de LTKB nos Experimentos 3 e 4: usando a abordagem <i>diferencial evolutiva multiobjetivo</i>	100
Tabela 11. Desempenho obtido em termos taxa de acerto das tarefas de recuperação e adaptação dos Experimentos 3 e 4: usando a abordagem <i>diferencial evolutiva multiobjetivo</i>	101
Tabela 12. Desempenho (em LTKB) de diferentes abordagens (trabalhos anteriores) quanto à economia de combustível (em LTKB): CBR+AG (CBR com <i>genética multiobjetivo</i>), CBR+ED (CBR com <i>evolução diferencial multiobjetivo</i>), CBR (CBR	

com algoritmo genético com soma ponderada), AM: (aprendizagem de máquina), DCOP (otimização de restrição distribuída), SE (sistema baseado em regras), SH (maquinista ser humano). 102

Tabela 13. Desempenho de diferentes abordagens (trabalhos anteriores) quando à economia de combustível (em %): CBR+AG (CBR com genética multiobjetivo), CBR+ED (CBR com evolução diferencial multiobjetivo), CBR (CBR com algoritmo genético com soma ponderada), AM: (aprendizagem de máquina), DCOP (otimização de restrição distribuída), SE (sistema baseado em regras), SH (maquinista ser humano). 103

Tabela 14. Desempenho obtido em termos taxa de acerto das tarefas de recuperação e adaptação dos Experimentos 1 e 2: comparativo das duas abordagens: *abordagem genética* (AG) e *abordagem diferencial evolutiva* (DE). 106

Lista de Gráficos

Gráfico 1. Taxa de acerto da tarefa de recuperação de casos: usando a abordagem <i>genética multiobjetivo</i> – Experimentos 1 e 2.	96
Gráfico 2. Taxa de acerto da tarefa de adaptação de casos: usando a abordagem <i>genética multiobjetivo</i> – Experimentos 1 e 2.	97
Gráfico 3. Taxa de acerto da tarefa de recuperação de casos: usando a abordagem <i>diferencial evolutiva multiobjetivo</i>	100
Gráfico 4. Taxa de acerto da tarefa de adaptação de casos: usando a abordagem <i>diferencial evolutiva multiobjetivo</i> – Experimentos 1 e 2.	101
Gráfico 5. Taxa de acerto da tarefa de recuperação de casos: usando as abordagens <i>genética multiobjetivo</i> (AG) e <i>diferencial evolutiva multiobjetivo</i> (DE) – Experimentos 1 e 2.	104
Gráfico 6. Taxa de acerto da tarefa de adaptação de casos: usando as abordagens <i>genética multiobjetivo</i> (AG) e <i>diferencial evolutiva multiobjetivo</i> (DE) – Experimentos 1 e 2.	105

Lista de Quadros

Quadro 1. Equações para os cálculos de resistência, tração, deslocamento, tempo e consumo. Fonte: (LOUMIET, JUNGBAUER e ABRAMS, 2005), (PROFILLIDIS, 2006), (AHMED A. SHABANA, 2007).....	28
Quadro 2. Visão consolidada dos trabalhos relacionados.....	69
Quadro 3. Representação de um caso.	79
Quadro 4. Condições de penalidade.	82
Quadro 5. Critérios de validação de uma solução.....	87

Lista de Abreviaturas

AG	<i>Algoritmo Genético/ Genetic Algorithm</i>
BDI	<i>Believe, Desire e Intention</i>
CBM	<i>Case Base Maintenance</i>
CBR	<i>Case-Based Reasoning</i>
DCOP	<i>Distributed Constraint Optimization</i>
DE	<i>Diferencial Evolutiva</i>
D-HS	<i>Discretised Highest Similarity</i>
ED	<i>Evolução Diferencial/Differential Evolution</i>
ENN	<i>Edited Nearest Neighbor Rules</i>
GPS	<i>Global Positioning System</i>
HS	<i>Harmony Search</i>
IGA	<i>Improved Genetic Algorithm</i>
IPM	<i>Método de Pontos</i>
KNN	<i>K-Nearest Neighbors</i>
LTKB	<i>Consumo de uma viagem (em litros por tonelada bruta transportada)</i>
MOP	<i>Multiobjective Optimization/Otimização Multiobjetivo</i>
NBI	<i>Normal Boundary Intersection</i>
NC	<i>Normal Constraint</i>
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm II</i>
PSO	<i>Particle Swarm Optimization (Otimização por Nuvem de Partículas)</i>
RAM	<i>Random Access Memory</i>
RBR	<i>Reasoning Rule-Based</i>
RENN	<i>Repeated Edited Nearest Neighbor</i>
RNN	<i>Reduced Nearest Neighbor Rule</i>
SNN	<i>Significant Nearest Neighbor</i>
SNN	<i>Selective Nearest Neighbor</i>
SOA	<i>Seeker Optimization Algorithm</i>
TER	<i>Threshold Error Reduction</i>
TER-S	<i>Simple Threshold Error Reduction</i>
TPMs	<i>Transition Probability Models</i>
VEGA	<i>Vector Evaluation Genetic Algorithm</i>

Sumário

CAPÍTULO 1	INTRODUÇÃO	18
1.1.	Motivação	19
1.2.	Problema de Pesquisa	23
1.3.	Objetivos	23
1.4.	Contribuições	24
1.5.	Estrutura da Tese	24
CAPÍTULO 2	FUNDAMENTAÇÃO TEÓRICA	25
2.1.	Ferrovias	25
2.1.1.	Locomotivas e Vagões	26
2.1.2.	Condução	27
2.1.3.	Ciclo de Condução	28
2.2.	Raciocínio Baseado em Casos	30
2.2.1.	Representação dos Casos e Aquisição de Conhecimento	33
2.2.2.	Recuperação de Casos	34
2.2.3.	Adaptação dos Casos	35
2.2.4.	Teste, Explicação, Reparação e Retenção dos Casos	37
2.2.5.	Manutenção da Base de Casos	37
2.3.	Otimização	41
2.4.	Otimização Multiobjetivo	41
2.5.	Conceitos Fundamentais de Otimização Multiobjetivo	42
2.6.	Métodos de Otimização Multiobjetivo	46
2.6.1.	Métodos a priori	46

2.6.2.	Métodos a posteriori	50
2.6.3.	Métodos Automáticos	51
2.6.4.	Discussão Métodos de Otimização Multiobjetivo	52
2.7.	Algoritmos Evolutivos	52
2.7.1.	Algoritmo Genético	53
2.7.2.	População Inicial.....	54
2.7.3.	Seleção	54
2.7.4.	Elitismo	55
2.7.5.	Mutação	55
2.7.6.	Cruzamento	55
2.8.	Algoritmo de Evolução Diferencial	56
2.8.1.	População Inicial.....	58
2.8.2.	Mutação	58
2.8.3.	Seleção	60
2.9.	Algoritmo Genético e Evolução Diferencial com Multiobjetivo.....	60
2.9.1.	Fast Non-Dominated Sorting	61
2.9.2.	Crowding Distance	62
2.10.	Trabalhos Relacionados	62
2.11.	Considerações do Capítulo	71
CAPÍTULO 3 MÉTODO PROPOSTO.....		73
3.1.	Domínio de Aplicação	74
3.2.	Ambiente de Condução.....	74
3.3.	Método	80
3.3.1.	Funções Objetivo	82

3.3.2.	Representação dos Indivíduos	83
3.3.3.	Mutação	83
3.3.4.	Cruzamento	85
3.3.5.	Seleção	85
3.3.6.	Adaptação dos Casos	86
3.3.7.	Revisão.....	86
3.3.8.	Método de Avaliação	88
3.4.	Considerações do Capítulo	88
CAPÍTULO 4 EXPERIMENTOS		90
4.1.	Ambiente de Execução dos Experimentos.....	91
4.2.	Cenários dos Experimentos	91
4.3.	Resultados Obtidos	93
4.3.1.	Abordagem genética multiobjetivo.....	94
4.3.2.	Abordagem diferencial evolutiva multiobjetivo	98
4.4.	Discussão dos Resultados	102
4.5.	Considerações do Capítulo	107
CONCLUSÃO E TRABALHOS FUTUROS.....		108
REFERÊNCIAS		111

Capítulo 1

Introdução

Ao longo dos últimos anos, vários estudos foram realizados no Laboratório de Pesquisa em Agentes de Software do Programa de Pós-Graduação em Informática da PUCPR. O marco inicial destes estudos foi o Projeto PAI-L (Piloto Automático Inteligente para Locomotivas), no qual foram realizados diferentes experimentos seguindo distintas abordagens para o desenvolvimento do piloto automático inteligente de locomotivas: aprendizagem de máquina (BORGES, 2009), programação por restrição (LEITE, 2009), programação reativa (DORDAL, 2010). Outro experimento foi realizado por (SILVA, 2011), o qual definiu um gerador de planos de condução de trens de alto nível baseado em uma arquitetura BDI (*Believe, Desire e Intention*). A partir de uma base já existente de planos de condução, surgiu um outro experimento proposto por (BORGES, 2015), usando raciocínio baseado em casos como método para reusar planos de condução; executando em particular as tarefas básicas de recuperação e adaptação de planos com algoritmo genético com soma ponderada. Ainda no contexto de condução automática de trens de carga (DORDAL, 2015) propôs uma arquitetura de agentes preditivos e autoconfiguráveis capaz de gerar e executar expectativas de ações em tempo de execução. Considerando que situações inesperadas podem ocorrer durante a condução de um trem de carga, o modelo computacional proposto antevia a situações de falhas nas ações que estariam programadas previamente e se autoconfigurava, visando a garantia da segurança, economia de recursos e economia de tempo. Esses experimentos individualizados, em termos abordagens diferentes, permitiram, de um lado, estabelecer resultados significativos para o domínio de aplicação em questão, e de outro lado, encorajar a realização de outros esforços de pesquisa, seja pela combinação de abordagens já avaliadas individualmente pelo grupo de pesquisa, seja pelo estudo de outras abordagens ainda não avaliadas para o problema em questão.

Considerando este cenário, e principalmente os trabalhos de (SATO, BORGES, *et al.*, 2012), de (BORGES, DORDAL, *et al.*, 2015) e de (BORGES, DORDAL, *et al.*, 2014), identificou-se algumas lacunas a serem preenchidas. Estas lacunas deram origem às seguintes hipóteses: (i) a utilização de algoritmo evolutivo—além de algoritmo genético—para realizar o processo de adaptação dos casos é viável; e (ii) a adoção de um método de otimização multiobjetivo pode apresentar melhores resultados.

Para esta tese mantem-se o foco de interesse a área de condução de trens, especificamente com esforços singulares na etapa de adaptação de planos de condução de trens interurbanos de carga. A tarefa de geração de planos de ações possui características de problemas multiobjetivo, à medida que ela deve considerar certas necessidades, tais como, assegurar: velocidade adequada às condições da via férrea, tempo gasto aceitável de viagem e consumo de combustível dentro de padrões competitivos. Esses objetivos são conflitantes, logo o método de geração de planos de ações deve assegurar um compromisso em termos de eficiência e eficácia. Eficiente de tal maneira que garanta que o trem seja conduzido de um ponto A até um ponto B com segurança, e eficaz de tal maneira que seja econômico em termos de consumo de combustível e tempo de duração de viagem.

1.1. Motivação

Ser capaz de utilizar experiências passadas é algo inerente ao ser humano, mas tirar o melhor proveito destas experiências nem sempre é possível, em particular, quando o processo de tomada de decisão requer rapidez. Esta situação ilustra o que ocorre na condução de trens por maquinistas, em que a pessoa do maquinista tem experiência para colocar em prática boas políticas de ações, mas dada a dinâmica das condições ambientais e de configurações (tipo de carga, tamanho de trem, distribuição da carga, condições da via, antecipação de uma tomada de ação, entre outras) manter regularmente boas políticas de ações não é trivial. Assim, a motivação para este projeto de pesquisa centra-se em obter melhor proveito das experiências passadas registradas em uma base de planos e assim orientar a atividade de condução, reaproveitando e/ou adaptando planos de condução para serem aplicados em situações presentes. Além de uma aplicação prática significativa, o problema abordado apresenta acentuada complexidade computacional, sendo assim justifica um estudo de abordagens que podem ser utilizadas para garantir a geração e experimentação—em banca de laboratório—de

políticas de ações eficazes e eficientes, por exemplo, em termos de otimização do tempo gasto em cada viagem, assim como do consumo de combustível e da velocidade do trem.

O consumo de combustível constitui a principal fonte de custo operacional para as empresas de transporte ferroviário. Segundo a Vale (Relatório Anual, 2013), em 2013 o custo com combustível no Brasil foi de 1.804 milhões de dólares, valor que representa 7,5% do custo de produção da empresa. Em 2014 (Relatório Anual, 2014), o custo com combustível no Brasil foi de 1.639 milhões de dólares, valor que representa 6,5% do custo de produção da empresa. A energia gasta—em termos de óleo combustível, diesel e eletricidade—é um componente significativo do custo de produção da Vale, representando 10,9% do custo total dos produtos vendidos em 2016 (Relatório Anual, 2016). Obviamente, estes custos afetam tanto o comércio interno quando as exportações e importações de produtos, uma vez que elas são repassadas ao consumidor final.

Além do consumo, outro fator importante é o tempo dispendido em uma viagem. O cenário ideal é realizar uma viagem no menor tempo possível com o menor consumo de combustível. Para alcançar o menor tempo se faz necessário aumentar a velocidade do trem. Todavia, aumentar a velocidade requer-se, na grande parte do tempo, fazer uso de maior potência, logo dispende mais recursos para gerar energia elevando os custos de LTKB. Neste cenário não trivial, aplicou-se a otimização multiobjetivo, tendo em vista otimizar o consumo de energia, o tempo de duração de viagem e a velocidade do trem.

A automatização da tarefa de condução pode ser vista como um problema complexo, à medida que há significativo número de variáveis de controle a monitorar e cálculos a efetuar em tempo real. Além disso, a seleção da potência a ser empregada para rebocar um trem de forma eficiente e eficaz requer uma iniciativa que vise otimizar o deslocamento do percurso. A escolha desta potência não é uma tarefa trivial, em particular, quando o potencial de força de tração é formado por diferentes locomotivas, que podem ou não estarem distribuídas ao longo do trem; a força resultante é formada pela combinação das potências das locomotivas. A complexidade é elevada devido a fatores como o volume de cálculos realizados e a dificuldade de antecipar ações face as condições ambientais do momento e do futuro próximo.

Na condução assistida, antes de selecionar uma potência a aplicar, deve-se conhecer uma série de informações, tais como: velocidade após a aplicação da ação, potência de frenagem, resistência inercial/normal/acidental de todas as locomotivas e vagões, força de tração máxima permitida, percentual de rampa, ângulo de curva, coeficiente de atrito, entre outros valores. Essa gama de informações permite calcular a resistência total de um trem em

oposição ao movimento. O processo para encontrar a potência que suplante tal resistência total é normalmente iterativo; deve-se frisar que não é permitido aplicar uma potência que gere patinação. Cada potência aplicada resulta em um novo conjunto de informações: força de aceleração, força de tração, consumo, tempo e deslocamento. Todas estas informações, geradas antes e após a seleção de uma ação, são calculadas várias vezes ao longo da viagem.

Para uma boa tomada de decisão, o condutor deve conhecer o trecho a percorrer além do seu ponto de visão local. Isto permite, de um lado, otimizar o uso da potência e do peso do trem a favor do movimento e, de outro, evitar a falta de potência e conseqüentemente parar o trem em determinado local diferente do seu objetivo final. Uma situação como esta pode resultar em problemas, tais como: (i) aumento do consumo de combustível devido à parada; (ii) congestionamento no trecho de via férrea; e (iii) necessidade de alocação de outra locomotiva para rebocar o trem (em caso extremo). Para evitar estes problemas, cada maquinista deve conhecer os pontos críticos do trecho de via férrea por onde está trafegando, de modo a prever corretamente as ações a realizar frente a cada situação crítica. Um sistema de condução baseado em recursos computacionais, no entanto, pode armazenar e utilizar um grande conjunto de informações sobre um trecho de via férrea e sobre ações executadas com sucesso em missões anteriores e reutilizá-las em missões futuras. Para acomodar tal reutilização de forma natural, o uso do ciclo canônico de raciocínio baseado em casos é peremptório.

O raciocínio baseado em casos pode ser visto como um método de aprendizagem incremental, sendo o desenvolvimento de uma tecnologia que usa a simulação computacional do pensamento e do raciocínio humano, em que todo o esforço feito em uma situação passada para resolver um problema é reaproveitado em uma nova situação. Outro ponto relevante é que a reutilização prévia de uma experiência não se restringe apenas ao sucesso passado como um guia para a obtenção de novo sucesso, à medida que é possível também reutilizar insucessos de forma a antever futuras situações de falhas e evitá-las. Assim, não há necessidade de usar recursos para corrigir esforços de potenciais falhas na situação atual (SCHANK e RIESBECK, 1989).

O processo canônico de raciocínio baseado em casos tem cinco etapas. A primeira etapa consiste, a partir da percepção (e.g., valores lidos de sensores) e de uma representação do domínio de aplicação, gerar uma formulação do problema-alvo (Formular). Dada a formulação do problema-alvo, a segunda etapa consiste em recuperar, a partir de uma memória de casos passados, casos relevantes para resolver o problema-alvo (Recuperar). Um caso é constituído de um problema, sua solução, e, em geral, as anotações sobre a maneira de como a solução foi

derivada. Por exemplo, o maquinista *Thor* quer manobrar um trem de carga com 60 vagões. Sendo ele um maquinista iniciante, a experiência mais relevante que ele pode recordar é aquela em que ele fez com sucesso, a saber: a manobra de trens em pátio de classificação (local no qual os vagões são separados e reagrupados em blocos segundo um destino comum, que pode ser o destino final da carga ou outro pátio). O procedimento que ele seguiu para fazer as manobras em pátio de classificação em conjunto com as justificativas para as decisões tomadas ao longo do tempo, constitui o caso recuperado por *Thor*. Depois de encontrar o caso mais similar, a próxima etapa consiste em mapear a solução do caso recuperado para o problema-alvo (Reusar). Isto pode envolver a adaptação da solução para adequá-la à nova situação; a adaptação é uma tarefa complexa e fundamental para a obtenção das soluções ao problema-alvo. No exemplo manobras de trens, *Thor* deve adaptar a solução que ele recuperou de modo a rebocar um trem de carga com vários vagões. Para tal, ele deve determinar qual é a diferença entre o problema-alvo e o caso recuperado, e modificar a solução recuperada para levar em conta essas diferenças. Tendo sido mapeada a solução recuperada para a situação-alvo, a próxima etapa consiste em testar a nova solução no mundo real (ou em uma simulação realista) e, se necessário, revê-la (Revisar). Supondo que *Thor* adaptou sua solução de manobra em pátio de classificação adicionando mais uma locomotiva ao trem. Após tal adição, ele descobre que a força de tração não é suficiente. Assim se faz necessário uma revisão. Nesta revisão *Thor* acrescentou uma locomotiva para cada 20 vagões do trem para ter a disposição à força de tração necessária para superar a resistência de inércia inicial. Quantas adaptações precisam ser feitas? Isto depende da natureza das diferenças entre o problema-alvo e o caso recuperado. Depois que a solução foi adaptada com sucesso ao problema-alvo, a próxima etapa, consiste em armazenar a experiência resultante como um novo caso na memória (Reter). *Thor* registra o procedimento que realizou para rebocar trens com 60 vagões, enriquecendo assim o seu conjunto de experiências armazenadas, e se preparando melhor para as futuras demandas de manobras de trens.

A abordagem de condução desenvolvida nesta tese considera diretamente as experiências de condutores humanos (adaptando-as se necessário), assim como as experiências de outros sistemas autônomos de condução. Nesta linha, fundamentada no raciocínio baseado em casos (AAMODT e PLAZA, 1994), cuja ideia principal é que, para solucionar um novo problema, uma entidade (pessoa ou software) não precisa criar um plano inteiramente novo passo a passo. Ao invés disso, ela relembra situações passadas similares, e as adapta de acordo com a nova situação. A solução do novo problema amplia a base de experiências. Tal

capacidade incremental online de aprendizado, mesmo em ambientes diferentes dos habituais, suaviza e facilita a aquisição e manutenção de conhecimentos (KOLODNER, 1993). Para este processo de adaptação foi utilizada a otimização multiobjetivo, buscando otimizar o tempo de duração de viagem, o consumo de combustível e a velocidade do trem.

1.2. Problema de Pesquisa

O problema de pesquisa pode ser definido como um problema de adaptação de casos, sendo estes casos, planos de condução de trens. Um plano de condução de trens descreve ações que devem ser executadas pelo maquinista em um determinado trajeto. Este plano deve considerar critérios previamente definidos, como por exemplo: velocidade máxima para o trajeto, tipo do trajeto (plano, aclive, declive), formação do trem (quantidade e pesos de vagões e locomotivas), entre outros. O plano também deve ser eficiente e eficaz. Eficiente porque deve consumir a menor quantidade possível de combustível, no menor espaço de tempo e com a maior velocidade possível (respeitando as normas de condução de trens interurbanos de carga e em particular as velocidades permitidas de cada via); e eficaz porque deve conseguir deslocar o trem de um ponto A até um ponto B, respeitando normas de segurança.

1.3. Objetivos

O objetivo geral desta tese é definir métodos de adaptação de casos com otimização multiobjetivo. A consecução desse objetivo passa pelos seguintes objetivos específicos:

- realizar um estudo sobre diferentes técnicas e métodos de otimização multiobjetivo;
- avaliar algoritmos de otimização multiobjetivo que permitam adaptar eficientemente planos de ações;
- gerar planos de ações eficientes para conduzir trens interurbanos de carga com otimização multiobjetivo, sem a intervenção humana;
- avaliar experimentalmente o desempenho de cada algoritmo multiobjetivo selecionado na tarefa de reparação e adaptação de planos de ações com diferentes configurações de trens e de vias férreas. Aqui, o desempenho referir-se, por

exemplo, a economia de combustível (*em LTKB*) e o tempo de duração de uma viagem de trem completa (*em minutos*); e

- avaliar experimentalmente o desempenho de cada algoritmo selecionado quanto a taxa de acerto na tarefa de recuperação e adaptação de planos. Aqui, o desempenho refere-se à qualidade de cada caso recuperado e de cada caso adaptado. Deve-se salientar que quanto maior for qualidade, menor será o esforço da próxima etapa no processo de geração de um plano de políticas de condução de trens interurbanos de carga.

1.4. Contribuições

Este trabalho traz como principais contribuições: (i) uma arquitetura que propicia o reaproveitamento, criação e compartilhamento de políticas de condução de trens de carga interurbanos; (ii) uma arquitetura que faz a indexação, recuperação, reuso e armazenamento de experiências passadas de condução de trens interurbanos de carga; (iii) os planos de condução gerados podem ser utilizados como base de conhecimentos e/ou treinamento para condutores de trens via simulação; (iv) um método de adaptação multiobjetivo de planos de ações para a condução de trens interurbanos de carga; (v) um método de geração de políticas de condução de trens interurbanos de carga; e (vi) um ambiente computacional de avaliação de técnicas e algoritmos multiobjetivo para um domínio realista—condução de trens interurbanos de carga.

1.5. Estrutura da Tese

A tese está organizado da seguinte maneira: o Capítulo 1 fornece uma introdução sobre o problema de pesquisa, objetivo e motivação econômica e técnica-computacional; o Capítulo 2 apresenta a fundamentação teórica sobre o domínio do problema—condução assistida de trens interurbanos de carga, raciocínio baseado em casos, otimização multiobjetivo, algoritmo genético e algoritmo evolução diferencial, o Capítulo 3 descreve o método para a solução do problema de pesquisa; o Capítulo 4 descreve os experimentos realizados bem como os resultados obtidos nesta pesquisa; e finalmente, o Capítulo 5 contém as conclusões e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo examina alguns conceitos fundamentais para o entendimento do problema de pesquisa que se encerra entorno do desafio de geração de políticas de ações que atendam critérios de otimização multiobjetivo. Tais conceitos serão examinados na seguinte ordem: a) Conceitos fundamentais sobre ferrovias; b) Raciocínio baseado em casos; c) Otimização, e otimização multiobjetivo e de métodos de otimização; d) algoritmos evolutivos—algoritmo genético e algoritmo de evolução diferencial.

2.1. Ferrovias

Uma ferrovia é um sistema de transporte que se utiliza de trilhos bidirecionais, mas que possui uma direção única no momento da condução do trem. Ela é composta por vários elementos estruturais, tais como: trilhos, dormentes e estações. Estes elementos dão suporte para a movimentação dos trens e viabilizam o gerenciamento daqueles que trafegam na malha férrea.

Uma via permanente é uma via férrea pela qual um trem é rebocado. Os elementos que compõem uma via férrea permanente são:

- Linha Principal. Ela atravessa pátios, interliga estações e fornece o meio por meio por onde trens são rebocados;
- Linha Secundária. É utilizada para viabilizar a manutenção da linha principal—podem haver várias secundárias, mas com tráfego reduzido;
- Desvio. É uma linha adjacente a linha principal. Ele é usado para as manobras de formação, ultrapassagem e cruzamento de trens;

- **Aparelho de Mudanças de Via.** É um dispositivo instalado nos trilhos. Ele auxilia na mudança de um trem da via principal para uma via adjacente e vice-versa;
- **Passagem de Nível.** É um cruzamento entre uma ferrovia e uma rodovia; e
- **Estação Férrea.** É um local em uma linha férrea onde o tráfego pode ser controlado e/ou bloqueado. Ela serve também para gerenciar o controle de tráfego de trens. Um trem só pode seguir viagem se obtiver a permissão da estação.

2.1.1. Locomotivas e Vagões

As locomotivas são veículos automotores que trafegam sobre trilhos de uma via férrea. Elas utilizam formas diversas de energia para produzir movimento e servem para rebocar vagões de passageiros ou de carga em uma via férrea. A maioria das locomotivas utilizam motor diesel-elétrico ou elétrico. Frequentemente, combina-se múltiplas locomotivas, e elas são conectadas eletricamente para permitir controlar sincronamente as funções de força e frenagem.

Geralmente as locomotivas são equipadas com um computador de bordo, e este é responsável pela leitura de diversos sensores distribuídos ao longo de um trem. Estes sensores tem a função de informar o maquinista a velocidade, pressão no encanamento dos freios, ponto de aceleração aplicado, posição da manopla de reverso (ela indica o sentido de deslocamento do trem, para frente ou para trás), uso do freio dinâmico, entre outras. Locomotivas também podem ser equipadas com rádios de comunicação para permitir trocar informações com a Central de Controle de Tráfego, e também com GPS (*Global Positioning System*); este último é utilizado para determinar a posição do trem em uma via férrea.

Os vagões são veículos destinados à acomodação e ao transporte de carga, e cada vagão possui uma especificidade. Tal especificidade é demarcada no próprio vagão e determina sua carga nominal, capacidade máxima e tara. Cada vagão também possui uma placa indicativa dos ajustes do nível de freio. Estes ajustes são configurados por um mecanismo denominado válvula tríplex, por onde é configurado a distribuição de ar no vagão durante uma frenagem.

2.1.2. *Condução*

A condução de um trem é feita por um maquinista. Ele a realiza basicamente aplicando uma força de aceleração ou uma força de frenagem. Esta aplicação é controlada por uma manopla. A manopla possui diferentes posições para a força de aceleração, diferentes posições para a força de frenagem e uma posição neutra. A quantidade de posições depende das características de cada locomotiva e do sistema de controle de veículo.

A aceleração é a variação da velocidade praticada por um trem. Esta variação pode ser positiva ou negativa. A aceleração gera esforços tratores para superar forças contrárias ou resistências para colocar um trem em movimento. A aceleração positiva deve ocorrer: a) quando a velocidade atual de um trem é menor que a velocidade máxima permitida para a via férrea; b) quando a velocidade atual for menor que a velocidade mínima necessária para rebocar um trem; ou c) quando a velocidade atual é decrescente, ou seja, o trem está em uma rampa com percentual positivo. Assim, para acelerar um trem é necessário empregar uma força de aceleração maior que a atual, aumentando assim a potência gerada pelo motor.

A frenagem de um trem é realizada para controlar a velocidade. Ela pode ser empregada para realizar uma redução parcial ou total da velocidade. Os trens são equipados com 3 freios:

- Freio automático ou pneumático – formado por um compressor de ar em cada locomotiva. O processo de frenagem inicia-se no compressor de ar da locomotiva, o ar sai por um encanamento que passa por todos os vagões, o que implica em um intervalo de tempo entre seu acionamento e sua efetivação, dado que o ar percorrerá a tubulação de todos os vagões. Este freio atua apenas nas rodas dos vagões.
- Freio dinâmico ou motor – é utilizado por meio da inversão de pontos de aceleração, geralmente para isto utiliza-se da manopla dos pontos de aceleração.
- Freio independente – é aquele realizado pelas rodas das locomotivas por meio da força aplicada em sapatas de freio.

2.1.3. Ciclo de Condução

O ciclo de condução, tanto humana quanto automatizada, baseia-se em diferentes etapas, sendo elas: concepção, percepção, atuação e retroalimentação. A concepção é uma característica humana que indica uma capacidade de compreensão, ou seja, elaborar conceitos. A percepção corresponde a recepção de estímulos sensoriais, e assim identificar, organizar e interpretar estes estímulos. A atuação é feita com base na interpretação dos estímulos, e assim uma ação é tomada. A retroalimentação é o procedimento de informar os demais envolvidos (outras pessoas, sistemas, entre outros) acerca do desempenho ou da ação executada.

A condução humana se caracteriza pela tomada de decisão ser exclusivamente do maquinista. Para isto ele faz uso de equipamentos embarcados e em suas percepções sensoriais.

A condução assistida se caracteriza por usar um programa de computador para auxiliar o maquinista em sua tomada de decisão, ou conduzir o trem de forma automática. Neste caso, a percepção é obtida a partir de sensores instalados na locomotiva, e estes sensores alimentam o computador de bordo. Somente a leitura destes sensores não é suficiente para o sistema de condução assistida selecionar ações corretas, sendo necessário diversos cálculos para—com base na situação atual—projetar ações futuras. Um sistema de condução assistida—por meio de diferentes cálculos—tem por finalidade verificar situações de excesso, evita-las e assim não permitir desperdício de recursos. **O Erro! Fonte de referência não encontrada.** Quadro 1 apresenta as equações para tais cálculos. De forma breve, pode-se dizer que o procedimento padrão consiste em aplicar uma força de tração maior que a resistência total de um trem para rebocá-lo.

Quadro 1. Equações para os cálculos de resistência, tração, deslocamento, tempo e consumo. Fonte: (LOUMIET, JUNGBAUER e ABRAMS, 2005), (PROFILLIDIS, 2006), (AHMED A. SHABANA, 2007).

EQUAÇÃO	FUNÇÃO	DESCRIÇÃO
$R_T = \sum_{i=0}^{n_l} (R_{nl} + R_{cl} + R_\gamma + R_i) + \sum_{i=0}^{n_v} (R_{nv} + R_{cv} + R_\gamma + R_i)$	$f_{Rt}(R_{nl}, R_{cl}, R_\gamma, R_i, R_{nv}, R_{cv})$ $f_{Rt}(T, ST, pos)^*$	Resistência total do trem (em kgf).
$R_{nl} = 1,3 + \frac{29}{w} + 0,03 \times v + \frac{0,0024 \times A \times v^2}{w \times n}$	$f_{Rn}(w, v, A, n)$	Resistência normal da locomotiva (em kgf).

$R_{nv} = 1.3 + \frac{29}{w} + 0,045 \times v + \frac{0,0024 \times A \times v^2}{w \times n}$	$f_{R_v}(w, v, A, n)$	Resistência normal dos vagões em kgf.
$R_{cl} = 0,2 + \frac{100}{R} \times (br + b + 3,8)$	$f_{R_l}(R, br, b)$	Resistência de curva da locomotiva (em kgf).
$R_{cv} = \frac{500 \times b}{R}$	$f_{R_{cv}}(b, R)$	Resistência de curva do veículo (em kgf).
$R_i = 10 \times i$	$f_{R_i}(i)$	Resistência de rampa do veículo (em kgf).
$R_y = 4 \times \frac{v_F^2 - v^2}{\ell}$	$f_{R_y}(v_F, v, \ell)$	Resistência inercial (em kgf)
$F_t = \frac{273.24 \times 0.82 \times HP}{v}$	$f_{F_t}(HP, v)$	Esforço trator (em kgf).
$F_{tm} = \frac{W \times f}{1 + (0,01 \times v)}$	$f_{F_{tm}}(W, f, v)$	Força tratora máxima (em kgf)
$F_{ac} = F_t - R_T$	$f_{F_{ac}}(F_{tm}, R_T)$	Força de aceleração disponível (em kgf).
$v_F = \sqrt{v^2 + \frac{(F_{ac} \times 20)}{(4 \times W)}}$	$f_{v_F}(v, F_{ac}, W)$	Velocidade final desejada para cada deslocamento de 20m (em km/h).
$\Delta \ell = 4 \times \frac{W \times (v_F^2 - v^2)}{F_{ac}}$	$f_{\Delta \ell}(W, v_F, v, F_{ac})$	Deslocamento (em metros).
$\Delta t = 7,2 \times \frac{\Delta \ell}{v_F + v}$	$f_{\Delta t}(\Delta \ell, v_F, v)$	Variação da duração da ação (em segundos).
$v_m = \frac{\Delta \ell}{\Delta t} \times 3,6$	$f_{v_m}(\Delta \ell, \Delta t)$	Velocidade média (em km/h).
$\Delta t_{final} = \sum_{k=1}^n \Delta t_k$	$f_{\Delta t_{final}}(\Delta t_k)$	Tempo gasto total da viagem (em segundos).
$\Delta \lambda_{final} = \sum_{k=1}^n \Delta \lambda_k$	$f_{\Delta \lambda_{final}}(\Delta \lambda_k)$	Deslocamento total da viagem (em metros).
$\gamma = F_{ac} \times \frac{g}{W}$	$f_{\gamma}(F_{ac}, g, W)$	Aceleração do trem.
$C = \frac{\Delta t}{60} \times cp$	$f_C(\Delta t, cp)$	Consumo nominal para um intervalo de tempo (em litros/minuto).
$LTKB = \frac{\sum C}{W \times \Delta \lambda_{final}}$	$f_{LGT}(\sum C, W, \Delta \lambda_{final})$	Consumo de uma viagem (em litros por tonelada bruta transportada-LTKB).
$F_{fd} = pad \times v_F \times 1417,475$	$f_{F_{fd}}(v_F, pad), se v \leq 38,6km/h$	Força de frenagem dinâmica realizada pelos motores da locomotiva (em kgf) em velocidade abaixo de 38.6km/h.
$F_{fd} = pad \times v_F^{-1.06} \times 921526,6$	$f_{F_{fd}}(v_F, pad), se v > 38,6km/h$	Força de frenagem dinâmica realizada pelos motores da locomotiva (em kgf) em velocidade acima de 38.6km/h.
$F_{fa} = ar \times P_{cl} \times ra \times ef \times ca \times 10$	$f_{F_{fa}}(ar, P_{cl}, ra, ef, ca)$	Força de frenagem automática realizada pelos vagões de um trem (em kgf).

$F_f = F_{fd} + F_{fa} - R_i$	$f_{Ff}(ar, P_{cl}, ra, ef, ca, v_F, pad, R_i)$	Força de frenagem total.
<p>Legenda:</p> <p><i>A</i>: área do veículo (em pés-quadrados) <i>ar</i>: área do êmbolo do cilindro de freio (em m²) <i>b</i>: bitola da linha (em metros) <i>br</i>: base rígida do veículo <i>ca</i>: coeficiente de atrito entre sapata de freio e roda <i>cp</i>: consumo de combustível em determinado ponto de aceleração (em litros) <i>ef</i>: eficiência da timoneira de freio (em %) <i>f</i>: coeficiente de aderência entre os trilhos e rodas <i>g</i>: aceleração da gravidade (em m/s²) <i>HP</i>: potência do veículo (em cavalo de força) <i>i</i>: porcentagem da inclinação da via a cada 100m <i>kgf</i>: quilograma força <i>ℓ</i>: deslocamento desejado (em metros) <i>n</i>: número de eixos</p> <p><i>n_l</i>: número de locomotivas <i>n_v</i>: número de vagões <i>pad</i>: ponto de frenagem dinâmica <i>P_{cl}</i>: pressão do cilindro de freio (em kPa) <i>pos</i>: posição da cabeça do trem <i>T</i> sobre a <i>ST</i>. <i>R</i>: raio da curva (em metros) <i>ra</i>: relação da alavanca da timoneira de freio <i>ST</i>: descrição de um trecho de via férrea onde o trem <i>T</i> está posicionado. <i>T</i>: descrição de um trem em termos de locomotivas e seus vagões. <i>v</i>: velocidade (em km/h) <i>v_F</i>: velocidade desejada (em km/h) <i>w</i>: peso do veículo (em toneladas) <i>W</i>: peso do trem (em toneladas)</p>		

* essa função adapta a aplicação da função $f_{Rt}(R_{nl}, R_{cl}, R_{\gamma}, R_i, R_{nv}, R_{cv})$

Deve-se notar que um maquinista conduz um trem, em particular, reusando totalmente ou parcialmente planos de ações. Estes planos denotam as experiências vividas e se traduzem em uma abordagem para resolução de problemas.

2.2. Raciocínio Baseado em Casos

Raciocínio baseado em casos é um paradigma para resolução de problemas que em muitos aspectos difere de forma fundamental de outros enfoques da Inteligência Artificial (AAMODT e PLAZA, 1994). O principal enfoque do raciocínio baseado em casos é resolver um novo problema lembrando um problema anterior similar, e assim reutilizar a informação e o conhecimento daquela situação (SCHANK e RIESBECK, 1989). O objetivo é recuperar experiências passadas (casos), adaptá-las para um problema-alvo, aplicar e registrar o resultado como uma nova experiência.

Esta prática de resolver problemas com base na experiência passada é utilizada rotineiramente, isto é, ela é nata no ser humano. Cada situação nova (problema) que se apresenta, a primeira coisa que se faz é recuperar experiências passadas similares, e com base nesta experiência projetar a solução. Alguns exemplos: quando é necessário deslocar-se de uma cidade A até uma cidade B, imediatamente pensa-se no melhor caminho, e este caminho é obtido pensando em uma experiência passada. Um mecânico quando vai efetuar um reparo em

um carro que apresenta algum defeito, normalmente com base nos problemas relatados, busca casos semelhantes e adota uma solução igual ou parecida.

Como já dito, esta maneira de raciocínio é baseada na reutilização de solução passada para solucionar um problema-alvo. O ciclo de vida do raciocínio baseado em casos (AAMODT e PLAZA, 1994) (SCHANK e RIESBECK, 1989) consiste basicamente em:

- coletar dados ou informações do problema-alvo;
- recuperar o caso mais semelhante registrado em uma memória de casos com base nos dados de entrada que descrevem o problema;
- adaptar o caso recuperado utilizando regras específicas;
- testar a solução proposta, se tal solução não for adequada deve-se analisar as explicações e as causas, e assim reparar a solução, gerando uma nova solução; e
- registrar a solução na memória de casos para reuso futuro, se tal solução foi aprovada nos testes.

A Figura 1 representa o ciclo canônico de um sistema de raciocínio baseado em casos. Ele começa com a representação do caso, que é uma descrição do problema a ser resolvido.

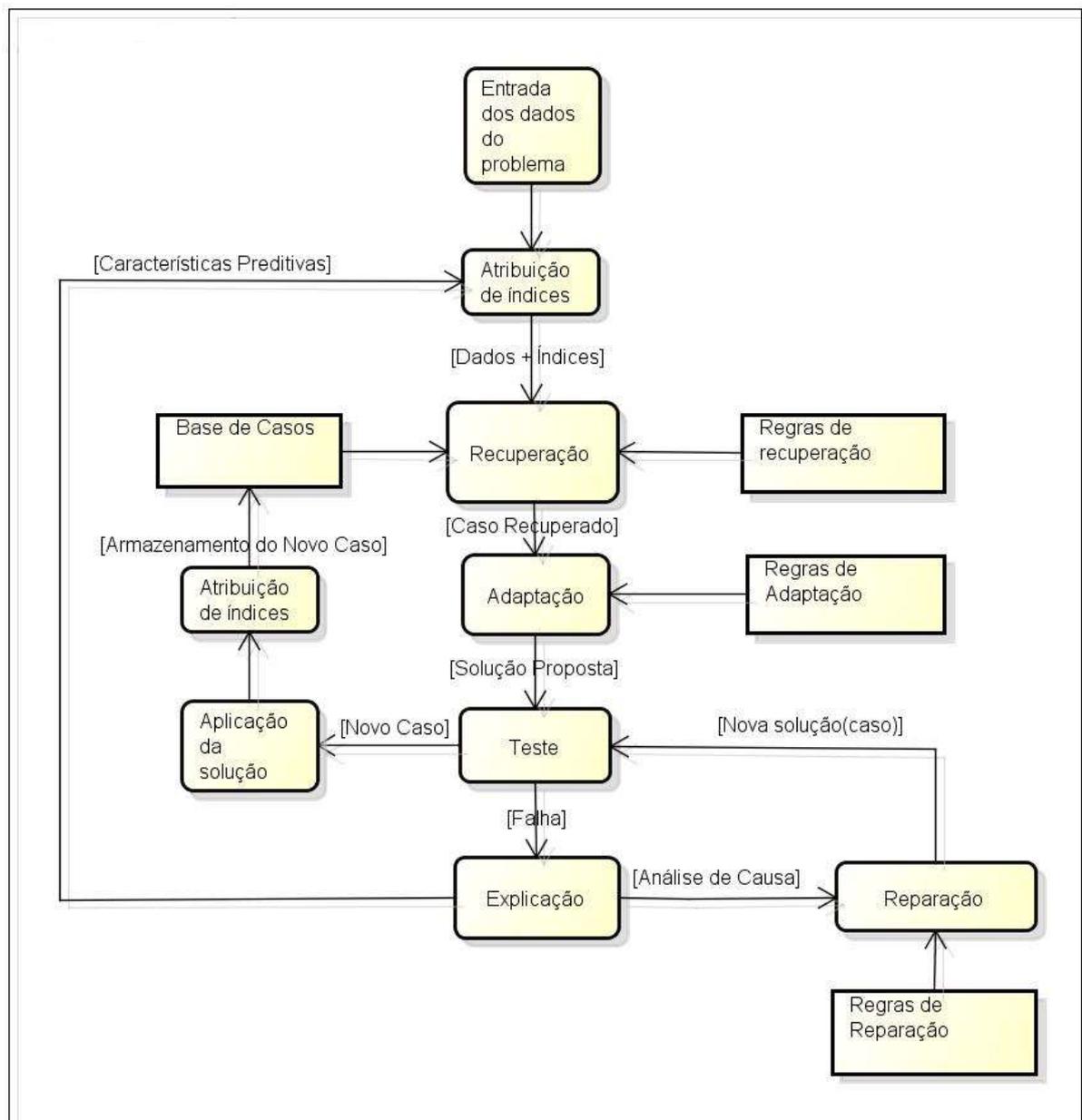


Figura 1. Ciclo de um sistema de raciocínio baseado em casos.

Fonte: (SCHANK e RIESBECK, 1989) adaptado.

A busca por um caso relevante envolve (SCHANK e RIESBECK, 1989): a identificação de características apropriadas para o problema-alvo (entrada), recuperação de casos da memória (de casos com tais características e escolha do(s) caso(s) que represente(m) da melhor maneira o problema-alvo.

Um sistema de raciocínio baseado em casos reúne alguns elementos básicos (WANGENHEIM e WANGENHEIM, 2003), que são:

- Representação do Conhecimento: é uma descrição de experiências concretas, normalmente de forma simbólica. Pode, se necessário, agregar outras formas de representação do conhecimento;
- Recuperação de Casos: consiste na tarefa de filtrar os casos relevantes para um problema-alvo, pode-se empregar medidas de similaridades para implementar tal tarefa, tais como: distância Euclidiana, distância *Manhattan*, entre outras;
- Adaptação de Casos: consiste na tarefa de adaptar um caso recuperado para atender o problema-alvo. Isto é necessário pois situações passadas representadas como casos dificilmente são idênticas aquelas do problema-alvo. Pode-se adaptar apenas valores de um caso ou até mesmo a sua estrutura; e
- Retenção de Casos: consiste na tarefa de armazenar a solução corrente ao problema-alvo, tal solução é um novo caso. Pode-se reter apenas valores de um caso em uma estrutura pré-definida ou, ir além, refazendo ligações entre casos.

2.2.1. Representação dos Casos e Aquisição de Conhecimento

O formalismo para a representação dos casos é de suma importância para garantir o sucesso de um sistema de raciocínio baseado em casos. Os casos podem ser representados de diversas formas, podendo-se utilizar quase todos os formalismos de representação do conhecimento, como frames (LIAO, MAO, *et al.*, 2012), representação de objetos (GRECH e MAIN, 2004), predicados, redes semânticas, redes neurais (SÁNCHEZA, COUSOB e CASILLA, 2009) e até a utilização do modelo relacional.

Definido o formalismo de representação, se faz necessário transformar os conhecimentos de um especialista de domínio em casos. Esta atividade é denominada aquisição de conhecimento. Ela pode ser feita de maneira manual, semiautomática ou automática (REZENDE, 2003). A aquisição de conhecimento manual é baseada em descrições acerca do conhecimento que pode ser obtida por meio de entrevistas. A semiautomática é baseada em teorias cognitivas ou em modelos já existentes. A automática emprega aprendizagem de máquina para induzir regras a partir de exemplos já cadastrados, ou mineração de dados (GUO,

HU e PENG, 2011), que a partir de uma considerável massa de dados tenta-se extrair regras, entre outras.

Um exemplo de aquisição de conhecimento automática é dado em (MANZOOR, ASIF, *et al.*, 2012). Ele utilizou algoritmo genético para fazer a geração da base de casos. Em (RAMÍK, SABOURIN e MADANI, 2013) propõe-se uma aquisição de conhecimento semiautomática, na qual um robô humanoide adquire conhecimento acerca do ambiente a partir da observação e da interação com um tutor ser humano. Em (BORGES, 2015) propõe-se, a partir de um log de ações, uma aquisição automática de conhecimento acerca da condução de trens interurbanos de carga; aqui caso é um plano de ações. Deve-se tomar nota que tal esforço será reusado em nossos experimentos.

A geração de uma base de casos importante requer esforços significativos, à medida que se faz necessário *extrair* o conhecimento de um especialista de domínio e *garantir* que este conhecimento corresponda à realidade, *formalizar* este conhecimento na forma de representação adequada, e *persisti-lo*.

2.2.2. Recuperação de Casos

Esta tarefa é considerada uma etapa crítica em um sistema de raciocínio baseado em casos, pois dada a descrição de um problema-alvo, é necessário colocar em prática um mecanismo para encontrar e retornar casos similares de uma dada base de casos. O mecanismo de busca deve ser eficiente, uma vez que a busca envolve comparar o quão semelhante um problema-alvo é de um ou mais casos. Esta comparação pode não ser simples, em particular, quando envolve o julgamento de valor e não apenas se um valor é igual a outro.

Para a recuperação de casos, pode-se empregar diferentes abordagens e na literatura tem-se um número considerável delas, como: utilização do algoritmo KNN (*K-Nearest Neighbors*) com distância Euclidiana e ponderação de valores de atributos (JIAO, PAN, *et al.*, 2013) (YAN, QIAN e ZHANG, 2014); modificação do algoritmo KNN para SNN (*Significant Nearest Neighbor*) (TSAI e CHIU, 2009); indexação de casos usando algoritmo D-HS (*Discretised-Highest Similarity*) e variações do algoritmo D-HS (GALUSHKA, PATTERSON e NUGENT, 2007); indexação por rotulação de casos (BRINKER e HÜLLERMEIER, 2007); indexação por *hashing* (MOHSIN, NORWAWI, *et al.*, 2010); recuperação utilizando algoritmo de colônia de formigas (MA, RU, *et al.*, 2009); recuperação utilizando busca com múltiplos

índices e múltiplos objetos (KLARIC, SCOTT e SHYU, 2012). Também verifica-se combinação de abordagens, como em (CHEN e JIANG, 2013) que combinou três abordagens para recuperação: sistema *fuzzy*, grau de semelhança multidimensional ponderada e árvore de decisão; e (KANG, KRISHNASWAMY e ZA, 2014) que combinou medidas de similaridade e regras de associação.

2.2.3. *Adaptação dos Casos*

Após o caso mais similar ser recuperado (item 2.2.2), normalmente, se faz necessário adaptá-lo antes de aplica-lo/reusá-lo. Esta etapa consiste em fazer alterações para aumentar a eficiência do caso recuperado. Adaptação é um processo difícil de projetar dentro de um *resolvedor* de problemas baseado em casos. Para vencer as dificuldades de adaptação de casos, tais como, a dependência de domínio, o custo computacional e a incapacidade de produzir novos casos para resolver problemas novos, diferentes técnicas e métodos de adaptação podem ser empregadas nesta etapa, como: algoritmos genéticos (GRECH e MAIN, 2004), redes neurais artificiais (ZHANG, LOUVIERIS e PETROU, 2007), entre outras.

Os métodos de adaptação podem ser classificados de acordo com o nível de exigência de conhecimentos do domínio em questão (MITRA e BASAK, 2005), e assim podem ser divididos em:

- Métodos de adaptação que exigem pouco conhecimento. Eles são independentes ou pouco dependentes do conhecimento do domínio. Por exemplo, usando um algoritmo genético, o espaço de representação pode ser guiado pelo conhecimento do domínio, mas a evolução não é necessariamente guiada pelo conhecimento de domínio. Assim, um algoritmo genético pode ser genericamente usado em muitos métodos de resolução de problemas. Estes métodos podem ser classificados em **métodos estocásticos** (algoritmos genéticos, redes bayesianas, entre outros) e **métodos determinísticos** (substituição baseada em adaptação e substituição derivacional).
- Métodos de adaptação que exigem grande volume de conhecimentos. Eles exigem muitos conhecimentos de domínio em comparação a métodos estocásticos e determinísticos. Podem ser classificados em três categorias: **Métodos de substituição**: certas partes dos casos armazenados são substituídos com os casos

ocorridos recentemente. Para determinar um substituto adequado, requer-se conhecimentos de domínio; **Métodos de transformação**: envolvem mudanças estruturais para a solução e, portanto, conhecimentos adequados de domínio são necessários; e **Ranking de casos recuperados**: todos os casos relevantes são tomados como entrada e com base na média ponderada de valores dos atributos, a adaptação é feita, isto é, casos recuperados são ordenados de acordo com uma medida de relevância calculada a partir da média ponderada. Para atribuir a ponderação adequada para cada atributo é necessário conhecimento de domínio.

Métodos de adaptação de casos normalmente incluem (LIAO, MAO, *et al.*, 2012):

- **Não-adaptação**: este método pode ser utilizado para os casos com condições complexas, mas com soluções simples; por exemplo, a resposta é sim ou não, ou apenas um simples julgamento é requerido para determinar categorização.
- **Adaptação manual**: este método é adotado na maioria das situações, pois como evita a automação, evita também as dificuldades técnicas de adaptação de caso. Pode ser utilizado para um processo em que a relação do caso com a solução seja complexa, ou que a solução seja complexa. Desta forma, a adaptação é feita por um especialista, após a recuperação dos casos mais similares.
- **Adaptação com raciocínio baseado em regras**: esta abordagem emprega regras e fórmulas (representando o conhecimento acumulado de especialistas da área), e por meio destas conclusões podem ser derivadas. Muitos trabalhos de pesquisa empregam esta abordagem. A função do raciocínio baseado em regras em casos existentes é estabelecer regras para orientar a adaptação (ROSENBLOOM e GOLDING, 1995) (PRENTZAS e HATZILYGEROUDIS, 2007) (SÁNCHEZA, COUSOB e CASILLA, 2009).

Exemplos de trabalhos que utilizam métodos de adaptação que exigem pouco conhecimento do domínio: (GRECH e MAIN, 2004) apresenta um sistema de raciocínio baseado em casos que faz uso de algoritmo genético no processo de adaptação dos casos. Como a qualidade de casos armazenados em uma base de casos tem um efeito significativo sobre as soluções resultantes, foi investigado o impacto da qualidade e quantidade de casos (população inicial do algoritmo genético) para adaptar soluções ajustadas para novos problemas; e (ZHANG, LOUVIERIS e PETROU, 2007) em um espaço de soluções multidimensional. Eles utilizaram *back propagation* para relacionar o espaço de soluções com as soluções.

Os esforços de pesquisa de (COJAN e LIEBER, 2010), a partir das descrições dos casos, buscam contradições entre os mesmos, isto é, dado um problema-alvo, busca-se casos semelhantes e identifica-se as contradições entre eles, e utilizando inferência dedutiva efetua-se as adaptações necessárias; este é um exemplo de trabalho que utiliza métodos de adaptação que exige grande conhecimento do domínio.

2.2.4. Teste, Explicação, Reparação e Retenção dos Casos

Após a execução da tarefa de adaptação de caso, o caso adaptado deve ser avaliado ou testado. Este teste deve validar se o caso pode ser aplicado, gerando assim—se válido—um novo caso. Este novo caso é aplicado e persistido na base de casos. Se o caso falhar no teste, então o caso precisa ser reparado. Para reparar um caso é necessário gerar uma explicação do porquê o caso falhou no teste, e a reparação deve ser feita a partir dos pontos de falha, utilizando para isto regras de reparação. Após a reparação, o caso é novamente testado, e se aprovado no teste, ele é aplicado e passa a fazer parte da base de casos como uma nova experiência válida; esse ciclo amplia no tempo a gama de soluções disponíveis.

2.2.5. Manutenção da Base de Casos

Alguns sistemas de raciocínio baseado em casos não fazem a manutenção dos casos. Logo, a base de casos pode ter um crescimento considerável e com o uso prolongado do sistema pode acarretar em dificuldades de recuperação dos casos. Estas dificuldades podem ser tanto em tempo de processamento, pela quantidade de casos armazenados, quanto pela quantidade de casos similares retornados. Em outras palavras, se todos os casos adaptados forem retidos, a base se tornará muito especialista e pouco generalista.

O sucesso de um sistema de raciocínio baseado em casos depende da qualidade das descrições dos casos e a eficiência do processo de recuperação, especialmente quando o número de casos é significativo. Para garantir essa qualidade, a manutenção do conteúdo de uma base de casos torna-se peremptória. Na literatura, há diversos trabalhos que propõem soluções para estes problemas citados.

Algoritmos de manutenção de bases de casos podem ser classificados em três classes ou estratégias (SMITI e ELOUEDI, 2011) (YAN, QIAN e ZHANG, 2014), a saber

- Estratégia de particionamento de base de casos;
- Estratégia de seleção para redução de base de casos; e
- Estratégia de otimização de base de casos—tradicionalis e baseadas na competência dos casos.

Nas estratégias de particionamento criam-se políticas de particionamento, e a partir desta política, constrói-se uma elaborada estrutura para a base de casos para facilitar a sua manutenção contínua. A política de particionamento cria uma coleção de caso distribuídos considerando algum critério, e como resultado do processo obtém-se um agrupamento. A partir de cada um deles constrói-se um caso representativo, o que leva a um subconjunto de atributos. Estas políticas permitem a adição e exclusão de casos em cada pequeno agrupamento sem o uso de toda a base de casos. Para efetuar este particionamento pode-se empregar algoritmos de árvores de decisão (SHIU , SUN, *et al.*, 2003), e algoritmos de agrupamento (SMITI e ELOUEDI, 2013), por exemplo. Este método é simples e fácil de empregar, à medida que ele decompõe uma grande base de casos em pequenos grupos. Porém, pode não ser adequado dependendo o nível de especialização necessária, em particular quando os agrupamentos podem se tornar numerosos e com poucos casos que os representem.

Estratégias de seleção para redução da base de casos visam reduzir um conjunto de dados, selecionando representantes significativos. Existem diversos métodos de redução de dados, como: CNN (*Condensed Nearest Neighbor Rule*) proposto por (CHOU, KUO e CHANG, 2006), RNN (*Reduced Nearest Neighbor Rule*) apresentado em (GATES , 1972), SNN (*Selective Nearest Neighbor Rule*) apresentado em (RITTER, WOODRUFF , *et al.*, 1975), ENN (*Edited Nearest Neighbor Rules*) apresentado em (WILSON, 1972), e RENN (*Repeated Edited Nearest Neighbor*) apresentado em (GUAN, YUAN, *et al.*, 2009). Estes métodos são baseados no algoritmo KNN (*K-Nearest Neighbor*) e adotam estratégias para exclusão de casos considerando os vizinhos mais próximos.

Estratégias de otimização de base de casos baseiam-se na política de apagar casos levando em consideração algum critério. Estas estratégias de otimização incluem métodos de exclusão tradicionais e métodos baseados em competência dos casos.

Os métodos de exclusão tradicionais são divididos em:

- Método de exclusão aleatória, em que toda a vez que a base de casos excede alguma limitação pré-definida um item aleatório é removido da base. Esta é uma

política de fácil utilização. Sua implementação pode ser simples, mas pode falhar, tendo em vista que se pode excluir casos relevantes.

- Método baseado na frequência de caso. Ele considera a frequência com que um caso é utilizado. No entanto, esta abordagem não evita a exclusão de casos importantes, à medida que não é necessariamente a frequência de uso de um caso que indica a sua importância.
- Método de Eliminação por utilidade, é baseada em métricas de utilidade de *Minton* (MINTON, 1990), escolhe-se um item da base de casos para exclusão estimando seus benefícios de desempenho. Este método de eliminação remove itens da base de casos com utilidade negativa.

O problema com ambas as abordagens (aleatória e frequência) é que os casos importantes podem ser excluídos por engano. Estas políticas de exclusão tradicionais podem ter resultados desastrosos para sistemas de raciocínio baseado em casos. A supressão dos casos críticos pode reduzir significativamente a competência de um sistema de raciocínio baseado em casos, tornando certas classes de problemas alvo permanentemente insolúveis.

Métodos baseados na competência da base de casos: várias abordagens estão focadas em preservar a competência da memória de casos por meio da eliminação de casos. Zhu (ZHU, 1998) definiu várias medidas de desempenho, por meio das quais se pode julgar a eficácia de uma base de casos, tais como: cobertura e acessibilidade. Seu método categoriza os casos de acordo com a sua competência em três categorias de casos (SMITI e ELOUEDI, 2011):

- Casos pivô: eles não são acessíveis por nenhum outro caso e suas eliminações reduzem diretamente as competências do sistema; eles são geralmente casos discrepantes.
- Casos de suporte: eles existem em grupos. Cada caso de suporte fornece uma cobertura semelhante a outros no grupo. A supressão de um caso no grupo não reduz competência. A supressão de todos de um grupo equivalente a exclusão de um caso pivô.
- Casos auxiliares: sua cobertura é definida como um subconjunto da cobertura de um de caso, assim ele não afeta a competência como um todo; sua supressão não afeta a base de casos.

Esta técnica de redução por competência elimina, primeiramente, casos auxiliares, depois casos suporte e por último, casos pivô. Esta abordagem é melhor do que as políticas de exclusão tradicionais para preservar a competência. No entanto, sua utilização não garante a preservação da competência da base de casos. O artigo intitulado “*Modelling the competence of case-bases*” cria uma estratégia híbrida de eliminação por utilidade. Em primeiro lugar, o método seleciona candidatos para a eliminação. Se houver apenas um candidato, então ele é excluído. Se houver um maior número de candidatos, seleciona-se o que tem a menor cobertura, e este é eliminado (SMYTH e MCKENNA, 1998). O artigo “*Informed case base maintenance: a complexity profiling approach*” propõe um método para identificação de redundância na base de casos. Este método identifica casos redundantes e potencialmente causadores de erros, e estes podem ser eliminados. A ferramenta de manutenção permite ao usuário experimentar e decidir quais casos serão eliminados (CRAW, MASSIE e WIRATUNG, 2006).

O trabalho intitulado de “*On dataset complexity for case base maintenance*” faz uma comparação de algoritmos para redução da base de casos, para tal foi criado um experimento no qual oito algoritmos de manutenção foram aplicados em bases de casos obtidas a partir de 25 bases de dados. Os algoritmos utilizados nestes experimentos foram: RENN (*Repeated Edited Nearest Neighbour*), BBNR (*Blame-Based Noise Reduction*), *Redundancy reduction phase of ICF*, *Redundancy reduction phase of RC (Reduction Cases)*, CRR (*Conservative Redundancy Reduction*), ICF (*Iterative Case Filtering*), RC (*Reduction Cases*), CBE (*Case-Base Editing*). Os resultados foram considerados provisórios. Porém, algumas conclusões foram obtidas, como: a redução de ruído utilizando o algoritmo RENN pode ser muito agressiva, pois muitas vezes apaga-se vários casos, e, por vezes, isto impacta a precisão. Os autores reiteram que se for importante eliminar muitos casos, pode ser melhor usar o ICFR ou RCR, à medida que eliminam menos casos que o RENN, aumentam a precisão. O algoritmo BBNR é mais eficiente na identificação de ruído do que o algoritmo RENN, porque ele exclui vários casos como o RENN, mas com menor impacto na precisão, sendo mais assertivo na escolha dos casos (CUMMINS e BRIDGE, 2011).

O trabalho “*When similar problems don’t have similar solutions*” apresenta um algoritmo que identifica e remove casos ruidosos com o auxílio de uma medida de complexidade local. Os resultados da avaliação confirmaram que o algoritmo foi superior aos algoritmos de referência na eliminação de casos ruidosos, como por exemplo: ENN (*Edited Nearest Neighbour*), RENN (*Repeated Edited Nearest Neighbour*), TER (*Threshold Error*

Reduction) e TER-S (*Simple Threshold Error Reduction*) (MASSIE, CRAW e WIRATUNGA, 2007).

O artigo “*Using clustering for maintaining case based*” faz a manutenção da base de casos usando *cluster*. Ele se baseia na ideia de que a análise de *cluster* para uma grande base de casos é a forma mais eficiente para construir novas bases de casos. Considerando cada *cluster* uma nova base de casos, estas bases são menores em tamanho. Isso propicia o uso de operações de manutenção mais simples. Este trabalho foi baseado na técnica de particionamento por agrupamento. As experiências mostraram a eficácia do método, tendo em vista critérios de desempenho da base de casos (SMITI e ELOUEDI, 2013).

O trabalho intitulado “*Memory and forgetting: An improved dynamic maintenance method for case-based reasoning*” propõe um método de manutenção dinâmica melhorada por memória seletiva que imita a função de memória seletiva do cérebro humano para salvar novos casos, e intencionalmente excluir (esquecer) casos antigos. As experiências mostraram a eficácia do método proposto. A memória seletiva e a política de esquecimento podem reduzir significativamente o tempo e espaço complexidade, e manter ou melhorar a precisão e desempenho do raciocínio baseado em casos (YAN, QIAN e ZHANG, 2014).

2.3. Otimização

Um processo de otimização consiste em extrair o melhor rendimento possível de algo, minimizando ou maximizando um ou mais objetivos. Este algo pode ser uma pessoa, uma máquina ou uma situação em que se deseja obter o melhor resultado. É um termo fortemente utilizado nas mais diversas áreas do conhecimento, como, por exemplo, na área de administração é comum ter como objetivo maximizar lucros, ou ainda, minimizar perdas.

Problemas que envolvam otimização são comuns no cotidiano, e a maioria deles podem ser considerados problemas de otimização multiobjetivo.

2.4. Otimização Multiobjetivo

O processo de tomada de decisão não é um processo trivial, pois normalmente envolve a análise de diversos fatores que influenciam a decisão final. Decisões geralmente são tomadas com base no senso comum ou na intuição, mas em algumas situações a tomada de decisão deve ser eficiente

e eficaz, não havendo espaço para erros. O processo de tomada de decisão pode ser considerado um problema de otimização multiobjetivo, à medida que normalmente estão envolvidos dois ou mais objetivos, e geralmente esses objetivos são conflitantes entre si. Objetivos conflitantes se caracterizam pelo sucesso de um implicar no insucesso de outro (COELLO, 1999). Em muitas situações são realizadas simplificações no problema original para combinar todos os objetivos em uma única função, ou escolhendo apenas os objetivos classificados como prioritários ou importantes.

Os problemas com as características supracitadas são considerados problemas multiobjetivo, e não apresentam uma solução única, apresentam de fato um conjunto de soluções válidas. Um exemplo de um problema multiobjetivo é a condução de um trem interurbano de carga, em que deseja-se conduzir o trem por uma via férrea (i) consumindo a menor quantidade possível de combustível, (ii) dispendendo o menor tempo possível, e (iii) trafegando na maior velocidade possível, respeitando as normas de segurança da via férrea. Como atender estes três objetivos, sendo que os mesmos são conflitantes, na medida que quanto menor o consumo de combustível, menor a velocidade e maior o tempo de viagem. Nesse tipo de situação é possível por meio da otimização multiobjetivo encontrar um conjunto de soluções que atendam os três objetivos, não necessariamente a solução ótima para os três objetivos, mas soluções que sejam um ponto de equilíbrio entre os três.

2.5. Conceitos Fundamentais de Otimização Multiobjetivo

Os problemas de otimização multiobjetivo consistem basicamente na obtenção de um conjunto de soluções que satisfaçam as condições previamente definidas acerca do problema que pretende-se resolver e minimize ou maximize uma função constituída por vários objetivos. Desta forma pode-se definir um problema de otimização multiobjetivo da seguinte forma (CASTRO, 2001) (DEB, 2001) (COELLO, 1999) (COELHO e ALOTTO, 2008):

Minimizar ou maximizar $z = (f_1(x), f_2(x), \dots, f_r(x))$. Sujeito a $x \in X^*$ em que:

$g(x)$: restrições;

X : espaço de decisões;

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

Z é a imagem de X ou espaço objetivo;

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

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

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

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

Definição 1. *Otimização de Multiobjetivo Geral*

Consiste em encontrar um vetor $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfaça as m restrições de desigualdade:

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

E as p restrições de igualdade:

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

E otimize o vetor função

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

Definição 2. *Ótimos de Pareto*

Um ponto $\vec{x}^* \in \Omega$ (Ω é a região factível) é um ponto Pareto-ótimo se para cada $\vec{x} \in \Omega$ $yI = [1, 2, \dots, k]$ tanto,

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

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

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

Esta definição diz que x é um ótimo de Pareto se não existe um vetor x factível que decemente algum critério sem causar um aumento simultâneo em pelo menos algum outro critério. O significado de ótimo de Pareto é considerado para todo o espaço das variáveis de

decisão, a menos que seja especificado o contrário, ou seja, um ponto A é uma solução ótima de Pareto e pertence à fronteira de Pareto se: não existe nenhum outro ponto B , tal que B domine A . Em geral, para cada região viável existe um subconjunto S de soluções não dominadas.

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

Definição 3. Dominância de Pareto

Um vetor $\vec{u} = (u_1, u_2, \dots, u_k)$ diz-se que domina outro vetor $\vec{v} = (v_1, v_2, \dots, v_k)$ se e somente se u é parcialmente menor que v , por exemplo, $\forall_i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\}: u_i < v_i$.

Definição 4. Conjunto de Ótimos de Pareto

Dado um problema de otimização multiobjetivo $\vec{f}(x)$, o conjunto de ótimos de Pareto (P^*) é definido como:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \vec{f}(x') \preceq \vec{f}(x)\} \quad (1)$$

Onde Ω é a região factível, e $\vec{f}(x')$ domina $\vec{f}(x)$.

Definição 5. Fronteira de Pareto

Para um problema de otimização multiobjetivo $\vec{f}(x)$ um conjunto de ótimos de Pareto P^* dado, a Fronteira de Pareto (PF^*) é definida como:

$$PF^* := \{\vec{u} = \vec{f} = (f_1(x), \dots, f_k(x)) \mid x \in P^*\} \quad (2)$$

Esta definição caminha na direção dos limites da região para o qual o problema foi proposto, i.e., nos pontos inferiores no plano das funções objetivo. A Figura 2 ilustra um exemplo de curva de Pareto para um problema de minimização de dois objetivos, sendo eles minimizar A e minimizar B . A solução ótima é aquela com menores valores para A e B e as

soluções não-dominadas são aquelas mais próximas de cada eixo do gráfico, por possuírem os menores valores dos critérios.

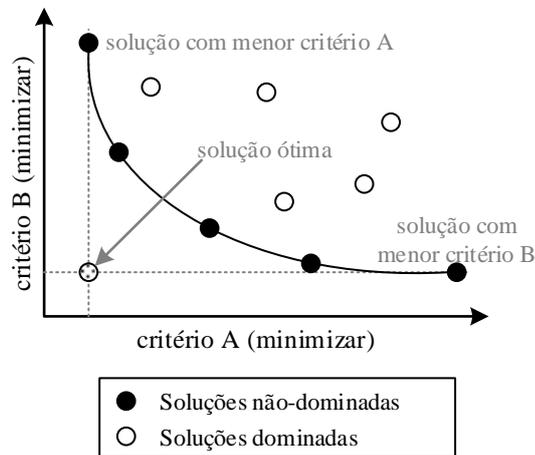


Figura 2. Curva de Pareto.
Fonte: (COELLO, LAMONT e VELDHUIZEN, 2007)

Os algoritmos evolutivos para resolver problemas multiobjetivo foram sugeridos no final dos anos 1960 por Rosenberg (ROSENBERG, 1967). A primeira implementação real de um algoritmo evolutivo multiobjetivo foi VEGA (*Vector Evaluation Genetic Algorithm*) (SCHAFFER, 1985), destinado a resolver problemas de aprendizagem de máquina. Desde então, uma ampla variedade de algoritmos foi proposta. Eles podem ser classificados em (ABRAHAM, JAIN e GOLDBERG, 2005):

- *Função de agregação.* Ela é considerada uma das abordagens mais simples para se trabalhar com um problema multiobjetivo. Consiste em combinar em uma única função todos os objetivos, utilizando para isto uma soma, multiplicação ou qualquer combinação aritmética de operadores.
- *Abordagem baseada em população.* Ela consiste na utilização de um algoritmo com base em uma população evolutiva para diversificar a busca. O exemplo clássico desse tipo de abordagem é o VEGA (SCHAFFER, 1985).
- *Abordagem baseada em Pareto.* Ela consiste em uma abordagem de seleção utilizando o conceito de ótimo de Pareto. Aqui, utiliza-se uma abordagem de base populacional, mas empregando na seleção dos indivíduos de uma população o conceito de ótimo de Pareto.

2.6. Métodos de Otimização Multiobjetivo

Os métodos de otimização multiobjetivo podem ser classificados da seguinte maneira: métodos que envolvem a articulação *a priori* de preferências. Isto requer que o usuário indique a importância relativa das funções objetivo ou o desejado de cada objetivo antes de executar a otimização. Métodos com uma articulação *a posteriori* de preferências, que envolvem a visualização por parte do usuário de conjunto de soluções matematicamente equivalentes, e o usuário escolhe uma destas soluções. Métodos que não exigem articulação das preferências, denominados métodos automáticos (MARLER e ARORA, 2004).

2.6.1. Métodos *a priori*

Os métodos *a priori* permitem ao usuário especificar preferências. Isto pode ser articulado em termos de metas ou de importâncias relativas dos diferentes objetivos a otimizar. A maioria desses métodos requerem parâmetros, que são coeficientes, expoentes, limites de restrições, entre outros, que podem ser definidos para refletir as preferências do usuário, ou serem continuamente alterados em um esforço para representar o ótimo de Pareto.

Existem diversos métodos *a priori*, a seguir são apresentados alguns deles:

1. Método critério global com ponderação. Ele é um dos métodos mais comuns para otimização multiobjetivo, no qual todas as funções objetivo são combinadas para formar uma única função. O peso do critério global (w) pode referir-se a qualquer função escalar, e é um tipo de função de utilidade em que os parâmetros do método são utilizados para modelar preferências. Uma das funções mais utilizadas geralmente é expressa em sua forma mais simples dada como a soma ponderada exponencial, conforme fórmula a seguir:

$$U = \sum_{i=1}^k [w_i F_i(x)]^p, F_i(x) > 0 \forall_i \quad (3)$$

2. Método da soma ponderada. Ele consiste também na combinação das funções objetivo em uma única função. Essa combinação é feita por meio de uma soma ponderada, conforme fórmula a seguir:

$$U = \sum_{i=1}^k w_i F_i(x) \quad (4)$$

3. Método lexicográfico. Ele dispõe as funções objetivo em ordem de importância. Em seguida, os problemas de otimização são resolvidos um de cada vez. Conforme fórmula a seguir:

$$\text{Minimize } F_i(x) \quad (5)$$

Sujeito a $F_j(x) \leq F_j(x_j^*)$, $j = 1, 2, 3, \dots, i-1$, $i > 1$, $i = 1, 2, \dots, k$.

Onde i representa a função de maior preferência.

4. Método *min-max* ponderado. Ele tenta minimizar a diferença entre $F_i(x)$ e F_i° . E w_i representa o peso pela qual a função objetivo é multiplicado. Conforme fórmula a seguir:

$$U = \max_i \{w_i [F_i(x) - F_i^\circ]\}. \quad (6)$$

5. Método exponencial ponderado. Ele foi proposto por (ATHAN e PAPALAMBRO, 1996) em resposta à incapacidade do método da soma ponderada capturar pontos em porções não-convexas de Pareto, conforme fórmula a seguir:

$$U = \sum_{i=1}^k (e^{pw_i} - 1) e^{pF_i(x)} \quad (7)$$

onde e é uma exponencial, p é uma constante e w_i é a ponderação atribuída a função objetivo e $F_i(x)$ é a função objetivo.

6. Método produto ponderado. Ele permite que as funções com diferentes ordens de grandeza tenham significado semelhante; isso evita a necessidade de transformar as funções objetivo, conforme fórmula a seguir:

$$U = \prod_{i=1}^k [F_i(x)]^{w_i} \quad (8)$$

Tal método é aplicado o produto entre as funções objetivo que representam o problema, neste caso, $F_i(x)$, e elas são elevadas a w_i que é o peso atribuído ao objetivo.

7. Programação com objetivos. Ela se baseia na especificação de n objetivos para cada função objetivo. Aqui, os objetivos b_j são especificados para cada função objetivo $F_j(x)$. Em seguida, o desvio total dos objetivos é minimizado, conforme fórmula a seguir:

$$\sum_{j=1}^k |d_j| \quad (9)$$

onde d_j é o desvio do objetivo b_j para o objetivo b_k . Para modelar os valores absolutos, d_j é dividido em partes positivas e negativas de tal forma que $d_j = d_j^+ - d_j^-$, com $d_j^+ \geq 0, d_j^- \geq 0$, e $d_j^+ d_j^- = 0$. O problema de otimização é formulado como segue:

$$\min_{x \in X, d^-, d^+} \sum_{i=1}^k (d_i^+ + d_i^-) \quad (10)$$

Sujeitos a:

$$F_j(x) + d_j^+ - d_j^- = b_j, j = 1, 2, \dots, k, \quad (11)$$

$$d_j^+, d_j^- \geq 0, j = 1, 2, \dots, k,$$

$$d_j^+ d_j^- = 0, j = 1, 2, \dots, k.$$

8. Método função objetivo delimitada. Ele visa minimizar um único objetivo; o mais importante da função objetivo. Todos os demais objetivos são transformados em restrições adicionais do problema.
9. *Physical Programming*. Inicialmente desenvolvido por (MESSAC, 1996), fornece um meio de adicionar preferencias sem ter que usar pesos relativos. Funções

objetivo, restrições e objetivos são tratados como métricas de projeto. Em geral, o usuário customiza uma função de utilidade individual, que é chamada $\bar{F}_i [F_i(x)]$, para cada métrica de projeto. Especificamente, cada tipo de métrica de projeto está associada inicialmente com um tipo de função utilidade individual. Em seguida, para cada métrica, o usuário especifica os intervalos numéricos que correspondem a diferentes graus de preferência (desejável, tolerável, indesejável, entre outros). Essas faixas incluem limites para os valores das métricas de projeto, que são modelados conforme restrições adicionais.

Dada a variedade de métodos existentes, a questão que surge é: qual é o melhor método? Não há uma resposta para esta pergunta. No entanto, os métodos apresentados proporcionam as condições necessárias e suficientes para obter ótimos de Pareto. Para quem está interessado em determinar uma solução única, as vantagens da obtenção de apenas soluções ótimas de Pareto são claras. Os métodos com capacidade de otimização são mais eficazes em refletir as preferências do usuário. Isto porque, assumindo que todos os pontos de Pareto são semelhantes matematicamente, distinguíveis apenas por preferências do usuário, não há razão em excluir soluções inerentemente potenciais. Tal exclusão pode suprimir do usuário uma solução que melhor reflete suas preferências.

Dos métodos que proporcionam uma condição necessária e suficiente para obter soluções ótimas de Pareto e refletir preferências do usuário, qual deve ser usado? A resposta para esta pergunta depende, em parte, sobre a precisão que cada um é capaz de se aproximar da função de preferência. *Physical programming* é eficaz a esse respeito, na medida em que ela permite customizar uma utilidade individual precisa para cada objetivo. Embora, opere com base em preferências impostas, ela proporciona um meio para contornar o uso de ponderação. No entanto, a codificação inicial pode ser relativamente complexa, pois é necessário conhecer em profundidade o problema para projetar as funções objetivo.

O estudo do método *physical programming* levanta uma questão central para problemas de otimização. Utilizando *physical programming* com multiobjetivo, o usuário precisa especificar uma quantidade relativamente grande de informações. Isto pode ser visto como um obstáculo ou como uma oportunidade. Um obstáculo à medida que nem sempre o usuário consegue especificar suas preferências, e em alguns problemas nem sempre as preferências são explícitas. É uma oportunidade porque a *physical programming* é relativamente eficaz em refletir preferências, e essa eficácia é uma consequência da capacidade

do método utilizar as informações fornecidas pelo usuário. Então, quanto mais informações o usuário fornecer, com mais precisão as preferências serão representadas.

Alguns métodos, como a soma ponderada, têm baixa capacidade de considerar informações de preferência, mas não são necessariamente prejudiciais, uma vez que podem existir casos em que as informações de preferência são limitadas ou simplesmente não existem.

2.6.2. *Métodos a posteriori*

Em alguns casos é difícil para o usuário expressar uma aproximação explícita da(s) função(ões) de preferência. Portanto, pode ser eficaz permitir que o usuário faça suas escolhas entre uma coleção de soluções. Para este fim, pode-se usar um algoritmo para determinar uma representação ótima de Pareto. Tais métodos usam uma articulação *a posteriori* das preferências (MESSAC e MATTSON, 2002).

Existem diversos métodos *a posteriori*, a seguir são apresentados alguns deles (MARLER e ARORA, 2004):

1. *Physical programming*. Apesar de ter sido inicialmente desenvolvido para a articulação *a priori* de preferências, ela pode ser eficaz no fornecimento de pontos ótimos de Pareto que representam com precisão o melhor conjunto completo de Pareto, mesmo quando a superfície ótima de Pareto não é convexa (MESSAC e MATTSON, 2002). De modo a representar um conjunto completo ótimo de Pareto para uma articulação *a posteriori* das preferências, (MESSAC e MATTSON, 2002), e (MESSAC, SUKAM e MELACHRINOUDI, 2001) fornecem um algoritmo para sistematicamente modificar as preferências como uma ferramenta matemática, em vez de uma indicação de preferências.
2. *Método NBI (Normal boundary intersection)*. Em resposta às deficiências na abordagem de soma ponderada, (DAS, 1999) propôs o NBI. Este método proporciona um meio para a obtenção de uma distribuição uniforme de Pareto. Para uma variação consistente usa-se o vetor escalar de parâmetros w , que é sistematicamente alterado para produzir uma distribuição uniforme de pontos ótimos de Pareto.

3. *Método NC (Normal constraint)*. Ele fornece uma alternativa ao método NBI com algumas melhorias (MESSAC, ISMAIL-YAHAYA e MATTSON, 2003). Quando usado com funções objetivo normalizadas e com um filtro de Pareto (que elimina soluções de Pareto não ótimas), esta abordagem fornece um conjunto de pontos ótimos de Pareto uniformemente espaçados. Nesses termos, ele sempre produz soluções ótimas de Pareto.
4. *Métodos Evolutivos*. São métodos baseados em comportamentos baseados na natureza, que simulam o processo de seleção natural e evolução das espécies, sendo que o mais famoso deste grupo é o genético.

Estes métodos permitem ao usuário visualizar opções (soluções) antes de tomar uma decisão. Eles não consideram qual a função objetivo é mais ou menos importante; só considera qual a solução mais atraente ou viável. Pode-se selecionar uma solução com base nos valores da função objetivo ou com base nos valores das variáveis de projeto. Eventualmente apresenta-se as soluções para o usuário em forma gráfica ou tabular.

As apresentações gráficas das soluções geralmente estão limitadas ao espaço tridimensional, e mesmo nas apresentações tridimensionais as representações de uma superfície de Pareto podem não ser claras. Quando apresentando soluções em forma de tabela, a seleção de uma única solução pode ser uma tarefa difícil considerando um grande número de objetivos, variáveis, ou pontos de solução. Conseqüentemente, estes métodos são os mais adequados para problemas com um número relativamente pequeno de objetivos.

2.6.3. *Métodos Automáticos*

Muitas vezes, o usuário não pode ou não consegue definir o que concretamente ele prefere. Esta seção descreve métodos que não exigem qualquer articulação de preferências. A maior parte dos métodos são simplificações de métodos com articulação *a priori*, tipicamente com exclusão de parâmetros.

Existem diversos métodos automáticos, a seguir são apresentados alguns deles:

1. *Método de Critérios Globais*. Ele tem seu fundamento na utilização de soma exponencial, produzindo uma única função objetivo. Porém, existem variações, como o método *min-max*, produto, soma, entre outros.
2. *Arbitragem Nash*. Trata-se de uma abordagem que é derivada da teoria de jogos. Com base em axiomas predeterminados de justiça sugere que a solução para um problema de arbitragem deve ser o máximo (em um conjunto compacto pontos) do produto de utilidades dos jogadores. Neste caso, as funções de utilidade sempre têm valores positivos e tem um valor zero na ausência de cooperação (quando nenhum acordo for alcançado). Em termos de formulação matemática, as funções objetivo individuais são minimizadas, e o método implica na maximização do critério global.

2.6.4. Discussão Métodos de Otimização Multiobjetivo

Em geral, a otimização multiobjetivo requer esforço computacional maior do que otimização com um único objetivo. Métodos com uma articulação *a priori* das preferências requerem do usuário a especificação das preferências. Alternativamente, métodos com uma articulação *a posteriori* de preferências permitem que o usuário visualize as possíveis soluções no espaço busca, e assim selecione uma solução aceitável.

Em termos de tempo de processamento, métodos com articulação de preferências *a posteriori* são menos eficientes do que métodos com uma articulação *a priori* das preferências. Dado que apenas uma solução é selecionada, o tempo gasto em determinar outros pontos ótimos de Pareto é desperdiçado (MARLER e ARORA, 2004).

2.7. Algoritmos Evolutivos

Algoritmos evolutivos são um subconjunto de algoritmos da computação evolutiva que implementam processos da evolução biológica como seleção natural, mutação e recombinação. Estes algoritmos possuem algumas características:

- uma base populacional;

- operadores de recombinação e mutação;
- processos de seleção para reprodução e sobrevivência.

Nas próximas sessões serão apresentados dois algoritmos evolutivos que foram utilizados nesta tese, sendo eles o algoritmo genético e algoritmo de evolução diferencial.

2.7.1. Algoritmo Genético

Simulações computacionais de sistemas genéticos tiveram início nos anos 50 e 60. John Holland começou a desenvolver as primeiras pesquisas sobre tema. Em 1975, Holland publicou o artigo intitulado "*Adaptation in Natural and Artificial Systems*", que foi considerado o ponto inicial dos algoritmos genéticos. David E. Goldberg, nos anos 80 obteve seu primeiro sucesso em aplicação industrial com algoritmo genético. Desde então tal iniciativa foi utilizada para solucionar problemas de otimização em diversas áreas do conhecimento.

Algoritmos genéticos simulam processos naturais de sobrevivência e reprodução de populações, essenciais para a evolução. Assim, como na natureza, indivíduos de uma mesma população competem em busca da sobrevivência e de conseguirem reproduzir-se, e assim passar seus genes para uma próxima geração. Desta maneira, os indivíduos mais aptos terão um maior número de descendentes. Para implementar um algoritmo genético é necessário (GOLDBERG, 1989) (COELLO, 1999) (DEB, 2001):

- Representações das possíveis soluções do problema no formato de um código genético;
- Definição de uma população inicial que contenha diversidade suficiente para permitir que um algoritmo combine características e produza novas soluções;
- Um mecanismo para medir a qualidade dos indivíduos;
- Um procedimento para combinar soluções e assim gerar novos indivíduos na população;
- Um critério de escolha dos indivíduos que permanecerão ou não na população;
- e
- Um mecanismo para introduzir periodicamente alterações em algumas soluções da população.

2.7.2. *População Inicial*

A representação de um indivíduo—em um algoritmo genético—é dada por meio do cromossomo. Um cromossomo é a estrutura que armazena informações e pode ser representado usando uma *string* de bits que recebe valores binários. Cada elemento da *string* é chamado de gene. Existem outros tipos de codificação além da codificação binária, como os citados por (GOLDBERG, 1989): codificação por valor, codificação por permutação, codificação em árvore.

A população inicial pode ser gerada de forma aleatória, ou se houver dados que possam dar origem a esta população inicial, mantendo as características de diversidade desta população, os mesmos podem ser aproveitados.

2.7.3. *Seleção*

É o método utilizado para selecionar quais indivíduos serão submetidos ao operador de cruzamento e também indivíduos que passarão para a próxima geração. Para tal pode-se utilizar a seleção através de torneio ou roleta (GOLDBERG, 1989).

- *Roleta*. Os indivíduos são copiados para a próxima geração de acordo com sua função de avaliação, uma probabilidade proporcional é calculada. A roleta é dividida em n partes, em que n corresponde a quantidade de indivíduos da população, e o tamanho de cada parte é proporcional à sua função de avaliação. A roleta gira por n vezes, e em cada giro um indivíduo é indicado, e este é inserido na nova população.
- *Torneio*. Um grupo de n indivíduos são obtidos aleatoriamente da população, podendo ser com reposição (o indivíduo selecionado, retorna e pode participar de outros torneios), ou sem reposição (o indivíduo selecionado não pode participar de outros torneios). Dentre n indivíduos que participaram do torneio é escolhido o indivíduo com melhor função de avaliação para a reprodução. Este processo se repete até que uma nova população seja gerada.

2.7.4. *Elitismo*

A cada geração, um indivíduo é modificado por meio de um mecanismo de cruzamento ou mutação, o que pode resultar na perda de soluções aceitáveis, ou até mesmo de uma solução ótima. Para evitar esta perda, surgiu a estratégia denominada elitismo. Esta estratégia garante através da função de avaliação que os melhores indivíduos da população permanecerão intactos para a próxima geração (GOLDBERG, 1989).

2.7.5. *Mutação*

O operador de mutação busca indivíduos aleatoriamente e altera o valor de alguns genes. Assim, um algoritmo genético cria novos cromossomos, gerando assim novas características que nem sempre estavam presentes na população inicial. Este processo ocorre de acordo com uma taxa de mutação previamente estipulada. Esta taxa é uma probabilidade que indica quais genes sofrerão a mutação (GOLDBERG, 1989).

2.7.6. *Cruzamento*

O operador de cruzamento permite criar novos indivíduos recombinação indivíduos da população. Para isso, são selecionados dois indivíduos (pais), que são divididos em uma determinada parte da *string* (esta escolha normalmente é aleatória), e então as informações são trocadas, gerando assim, dois novos indivíduos (filhos). A *string* pode ser quebrada em um ou n pontos, logo, o cruzamento pode ser de um ou n pontos (GOLDBERG, 1989).

Algoritmo 1. Algoritmo genético.

01	t ← 1;
02	Geração da população inicial
03	Avaliação
04	while (t < x) //critério de parada não atendido, x é a quantidade de gerações
05	for (i = 1 até N)
06	Avaliação
07	Atribuição de aptidão
08	End
09	Cruzamento
10	Mutaç�o
11	Aplicação de seleç�o
12	t← t+1;
13	End

2.8. Algoritmo de Evoluç o Diferencial

O algoritmo de evoluç o diferencial surgiu como uma forma competitiva da computaç o evolutiva. O primeiro artigo escrito sobre o tema apareceu como um relat rio t cnico escrito por Storn e Price em 1995 (STORN e PRICE, 1995). Um ano depois, o sucesso do algoritmo foi mostrado no *First International Contest on Evolutionary Optimization* (1 . ICEO) em maio de 1996, que ocorreu em conjunto com *IEEE International Conference on Evolutionary Computation* (CEC). O algoritmo em quest o terminou em terceiro lugar no 1 . ICEO, que foi realizada em Nagoya, Jap o. Ele acabou por ser o melhor algoritmo evolutivo para resolver um conjunto de funç es, e os dois primeiros lugares foram dados para algoritmos n o evolutivos, que n o s o universalmente aplic veis, mas resolveram o problema teste com maior rapidez do que o algoritmo evoluç o diferencial (DAS e SUGANTHAN, 2011).

Entre as principais caracter sticas do algoritmo de evoluç o diferencial destacam-se (DAS e SUGANTHAN, 2011):

- Uso de mecanismo de busca baseado no operador de mutaç o diferencial;
-   classificado como um algoritmo evolutivo, embora a mutaç o diferencial n o tenha base ou inspiraç o em nenhum processo natural; e
- Possui qualidades computacionais interessantes, como a simplicidade de implementa o, a robustez, a efici ncia, a auto adapta o e a versatilidade.

O uso do algoritmo de evoluç o diferencial tem sido visto como uma ferramenta de otimiza o atraente por muitos pesquisadores (DAS e SUGANTHAN, 2011). Em compara o

com a maioria dos algoritmos evolutivos, o algoritmo de evolução diferencial é muito mais simples de implementar. O corpo principal do algoritmo tem de 4 a 5 linhas de código fonte em quase todas as linguagens de programação. Nota-se que embora PSO (*particle swarm optimization*) é também fácil de implementar, o desempenho do algoritmo evolução diferencial e suas variantes é na maioria das vezes melhor que do PSO e suas variantes em uma grande variedade de problemas, como foi indicado nos estudos de: (DAS, ABRAHAM e KONAR, 2009), (RAHNAMAYAN, TIZHOOSH e SALAMA, 2008) e (VESTERSTROM e THOMSEN, 2004).

Segundo (STORN e PRICE, 1995) as principais vantagens do algoritmo de evolução diferencial são:

- Efetiva capacidade de otimização global;
- Operadores em formato de ponto flutuante com alta precisão;
- Algoritmo eficiente, por ser relativamente simples;
- Operação de mutação auto referencial;
- Otimização eficaz de parâmetros sobre inteiros e discretos;
- Opera em superfícies planas; e
- Pode fornecer várias soluções e é eficaz em problemas de otimização de restrição não-linear com funções de penalidade.

Como um membro da família dos algoritmos evolutivos, o algoritmo de evolução diferencial, também conta com a geração de uma população inicial, operador de mutação, recombinação e seleção por meio das gerações repetidas até que o critério de parada seja atendido. Seu funcionamento básico pode ser visto na Figura 3:

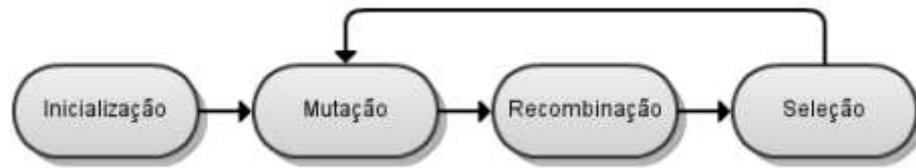


Figura 3. Funcionamento algoritmo de evoluçao diferencial.

2.8.1. Populaçao Inicial

Na geraçao G , a populaçao P^G é composto por $x_i^G, i = 1, 2, \dots, N$. A populaçao inicial P^{G0} pode ser escolhida aleatoriamente com uma distribuçao de probabilidade uniforme, cf.:

$$x_i^G = x_{i(L)} + rand_i[0,1] \cdot (x_{i(H)} - x_{i(L)}) \quad (12)$$

onde $x_{i(L)}$ e $x_{i(H)}$ são os limites inferiores e superiores de dimensões do vetor $x_i = \{x_{j,i}\} = \{x_{1,i}, x_{2,i}, \dots, x_{d,i}\}^T$. Porém, assim como no algoritmo genético, se houver dados para garantir a diversidade da populaçao, os mesmos podem ser aproveitados como populaçao inicial.

2.8.2. Mutaçao

O mecanismo de busca do algoritmo de evoluçao diferencial utiliza vetores-diferença criados a partir de pares de vetores da própria populaçao, com o seguinte procedimento: dois indivíduos são selecionados aleatoriamente da populaçao da geraçao atual, criando-se um vetor-diferença que dado pela diferença entre estes dois indivíduos; este vetor-diferença, por sua vez, é somado a um terceiro indivíduo, também selecionado aleatoriamente, produzindo uma nova soluçao mutante; a nova soluçao mutante é portanto o resultado de uma perturbaçao em algum indivíduo da populaçao, sendo esta perturbaçao um vetor-diferença construído aleatoriamente.

Conseqüentemente, as diferenças dos vetores-diferença passam para as funções objetivo informações topográficas para o processo de otimizaçao, e, portanto, fornecer maior capacidade de otimizaçao global e eficiência (WONG e DONG, 2005).

A equação a seguir representa tal procedimento:

$$x' = x_{r3}^G + F \cdot (x_{r1}^G - x_{r2}^G) \quad (13)$$

onde x' representa a solução mutante e F é um fator de escala aplicado ao vetor-diferença (constante de mutação) e parâmetro do algoritmo de evolução diferencial. O vetor x_{r3}^G , ao qual é aplicada a mutação diferencial, é denominado vetor de base. Usando este procedimento, obtém-se uma população mutante: $V_t = \{v_{t,i}, i = 1, \dots, N\}$.

A recombinação é obtida da seguinte maneira: os indivíduos da população atual P^G são recombinados com os indivíduos da população mutante V , produzindo a descendência ou população de soluções teste U^G . Na versão clássica do algoritmo de evolução diferencial, emprega-se a recombinação discreta com probabilidade $C \in [0,1]$, onde:

$$U_{t,i,j} = \begin{cases} v_{t,i,j} & \text{se } U_{[0,1]} \leq C \text{ } V_j = \delta_i \\ x_{t,i,j} & \text{caso contrário} \end{cases} \quad (14)$$

em que $\delta_i \in \{1,2, \dots, N\}$ é um índice aleatório obtido para o vetor teste i . Como em algum momento a igualdade $j = \delta_i$ é verificada, essa condição garante que pelo menos um dos parâmetros da solução teste é herdado do indivíduo mutante. O parâmetro C controla a fração de valores em $u_{t,i}$ que são copiados do vetor mutante $v_{t,i}$. O Algoritmo 2 apresenta como este processo acontece.

Algoritmo 2 – Evolução diferencial.

01	$t \leftarrow 1;$
02	Inicializar população
03	while ($t < x$) //critério de parada não atendido, x é a quantidade de gerações
04	for ($i = 1$ até N)
05	Selecionar vetores de base $x_{t,base}$
06	Selecionar conjunto de vetores-diferença
07	Gerar solução mutante
08	Adicionar $v_{t,i}$ a população mutante V_t
09	End
10	Recombinar x_t e v_t com parâmetro C usando V_t
11	Aplicar seleção
12	$t \leftarrow t + 1;$
13	End

2.8.3. Seleção

O processo de seleção dos indivíduos que passarão para a próxima geração é feito através da função de avaliação ou *fitness*. Assim, operador de seleção visa simplesmente escolher, dentre a população corrente e a população cruzada, os melhores indivíduos. É uma verificação simples. Se a *fitness* do indivíduo i da população é maior do que a *fitness* do indivíduo i da população cruzada, esse indivíduo passa para a próxima geração. A próxima geração nada mais é do que uma população com os melhores indivíduos da iteração corrente, logo, são os mais aptos.

2.9. Algoritmo Genético e Evolução Diferencial com Multiobjetivo

Quando se trabalha com múltiplos objetivos é importante o conceito de dominância para determinar a estratégia a ser adotada na classificação e seleção dos indivíduos da população. O critério de dominância é dado da seguinte forma: sendo dois indivíduos p e q de uma população P . Um indivíduo p domina um indivíduo q se no mínimo um valor em um dos objetivos de p é melhor que o mesmo objetivo em q e o restante dos valores dos objetivos de p não podem ser piores que o restante dos mesmos valores nos objetivos em q . Logo, p não pode ter nenhum objetivo com menor qualidade que os objetivos de q . Após esta análise, os indivíduos são classificados, i.e., eles são atribuídos a uma fronteira. As fronteiras são:

- *Front 1*. Todas as soluções não dominadas são consideradas as melhores soluções;
- *Front 2*. Todas as soluções não dominadas, excluídas as soluções incluídas no *Front 1*;
- *Front 3*. Determinado por meio da exclusão das soluções presentes no *Front 1*, e no *Front 2*.

Para a realização da seleção dos indivíduos foram utilizados dois mecanismos, a saber: *Fast Non-Dominated Sorting* e a *Crowding Distance*.

2.9.1. *Fast Non-Dominated Sorting*

O processo realizado pelo *Fast Non-Dominated Sorting* é realizado em 2 etapas. O Algoritmo 3 implementa a etapa de classificação dos indivíduos (*Etapa 1*) e o Algoritmo 4 implementa a etapa de inclusão dos indivíduos em um *front* (*Etapa 2*).

Algoritmo 3. Etapa 1: classificação dos indivíduos.

P : População; p, q : indivíduos da população; np : número de indivíduos que dominam p ; S_p : lista de indivíduos dominados por p ; F_1 : <i>front</i> 1.	
01 02 03 04 05 06 07 08	para todo $p \in P$ para todo $q \in P$ se p domina q então S_p recebe q senão se q domina p então $np = np + 1$ Se $np = 0$ então F_1 recebe p

A Etapa 1 analisará todos os indivíduos da população P , comparando uns com os outros e classificando-os. Ao final da Etapa 1, se o indivíduo possuir np igual a zero, significa que este indivíduo não é dominado por nenhum outro indivíduo da população P , e este indivíduo fará parte do *Front* 1.

A Etapa 2 classifica cada indivíduo em seu respectivo *front* de acordo com seus valores de dominância, indicados pelos seus valores de np . Cada indivíduo que é incluído em um *front* é retirado da população P , decrementando os valores de nq de cada indivíduo dominado por esses. Este processo se repete até que não reste mais nenhum indivíduo na população P .

O contador i controla a quantidade de *fronts*, a lista H armazena temporariamente os *fronts* gerados, tanto que ao final da etapa, F_i recebe H , e este é inicializado vazio. Um indivíduo p do *front* F_i é selecionado, e uma verificação é realizada para encontrar quais são os indivíduos q dominados por p (lista S_p). Os valores de nq destes indivíduos são decrementados com o objetivo de retirar do contexto os indivíduos já classificados no *front* anterior. A intenção aqui é fazer que indivíduos com nq igual a zero não façam parte do próximo *front*.

Algoritmo 4. Etapa 2: inclusão dos indivíduos em um *front*.

<i>i</i> : contador da quantidade de <i>fronts</i> ; <i>p</i> : indivíduo selecionado do <i>front</i> F_i ; <i>q</i> : indivíduo selecionado da lista S_p ; <i>nq</i> : número de indivíduos que dominam <i>q</i> ; S_p : lista de indivíduos dominados por <i>p</i> ; F_i : <i>front</i> atual; <i>H</i> : armazena temporariamente os próximos <i>fronts</i> , é inicializado vazio.	
01 02 03 04 05 06 07 08 09 10	$i = 1$ enquanto $F_i \neq \emptyset$ $H = \emptyset$ para todo $p \in F_1$ para todo $q \in S_p$ $nq = nq - 1$ Se $nq = 0$ então <i>H</i> recebe <i>q</i> $i = i + 1$ F_i recebe <i>H</i>

2.9.2. Crowding Distance

O *Crowding Distance* é um operador de diversidade. O objetivo é garantir uma melhor distribuição das soluções ao longo da linha de Pareto. Este operador calcula a distância média entre um ponto central *i* selecionado dentro da população e dois pontos localizados nas extremidades do ponto central ($i - 1$) e ($i + 1$). A intenção é que a partir de um ponto central, o operador de diversidade consiga encontrar extremos e priorizar pontos mais distantes durante o processo de seleção, e com isso espalhar resultados ao longo da linha de Pareto. As disposições dos pontos extremos formam um cuboide em relação ao ponto central.

2.10. Trabalhos Relacionados

Na literatura foi feita uma pesquisa de trabalhos que possuíam alguma semelhança esta tese. Para esta pesquisa foram utilizadas as palavras chave: CBR, algoritmo de evolução diferencial, algoritmo genético e otimização multiobjetivo. Estas palavras chave foram parametrizadas da seguinte maneira para efetuar a busca:

- CBR E algoritmo genético; OU
- CBR E algoritmo de evolução diferencial; OU
- algoritmo genético E otimização multiobjetivo; OU

- algoritmo de evolução diferencial E otimização multiobjetivo.

A busca foi realizada na base de periódicos da Capes e resultou em 154 artigos. Esta busca foi refinada a partir da leitura dos resumos de cada trabalho, e destes foram selecionados os que foram considerados mais semelhantes. Estes trabalhos são apresentados a seguir.

O artigo intitulado “*Integrated Short-Haul Airline Crew Scheduling Using Multiobjective Optimization Genetic Algorithms*”, propõe uma alternativa evolucionária para métodos de planejamento para resolver o problema de agendamento de tripulação. O problema se apresenta como uma otimização combinatória com várias restrições e objetivos. O esforço foi considerar simultaneamente a disponibilidade da tripulação e os turnos da tripulação. Para melhorar a eficiência do conjunto Pareto, a metodologia aplicou uma variação do algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*), que é um dos mais populares algoritmos evolucionários de otimização multiobjetivo. A abordagem proposta foi avaliada em um estudo de caso de uma tripulação de companhia aérea real. Os resultados experimentais obtidos foram comparados com um plano real da companhia aérea gerado pelo método convencional. Os resultados da comparação confirmaram que a variante proposta obtém soluções que são superiores em dois aspectos: primeiro, a variante NSGA-II obtém melhor convergência nos problemas estudados em comparação com a versão original do algoritmo; segundo, os resultados explorados pela variante NSGA-II permitem aos gestores selecionar várias agendas da tripulação, que são superiores ao plano real gerado pelo método convencional (CHEN, LIU e CHOU, 2013).

O artigo “*Multiobjective Electromagnetic Optimization Based on a Nondominated Sorting Genetic Approach With a Chaotic Crossover Operator*” Propõem uma modificação no algoritmo NSGA-II, a modificação proposta consistiu no uso de sequências caóticas baseado no mapa Zaslavskii (ZASLAVSKY, 1978), no operador de cruzamento. O algoritmo foi testado no problema de uma oficina que consiste em determinar um design otimizado de um armazenamento de energia magnética supercondutora. O objetivo foi armazenar uma quantidade significativa de energia em campos magnéticos com uma bobina bastante simples e encontrar os múltiplos pontos de ótimos de Pareto considerando duas funções objetivo: o campo de dispersão (para ser minimizada) e um nível de energia a ser alcançado. Os resultados foram comparados a solução clássica do NSGA-II. Os resultados das simulações mostraram que algumas das soluções não-dominadas obtidas pela solução clássica do NSGA - II são dominadas pela modificação NSGA-II. Além disso, a modificação NSGA-II obteve uma distribuição

significativamente melhor do que aquela obtida pela clássica NSGA-II, permitindo, assim, o fabricante ter mais opções para a tomada de decisão (COELHO e ALOTTO, 2008).

O artigo “*Multi-objective evaluation based optimizing control method for shaft furnace roasting process*”, propõe uma abordagem para controlar fornos utilizados no processamento de minérios de ferro. Este processo é considerado complexo e difícil. Para lidar com este problema, uma abordagem de otimização multiobjetivo foi desenvolvida por meio da combinação de teoria de avaliação com raciocínio baseado em casos. O conjunto de pontos de controle são produzidos pela metodologia de raciocínio baseado em casos, seguida pela avaliação em tempo real e correção do objetivo integrado com os índices de produção. Nos testes realizados conclui-se que todo o sistema apresentou maior confiabilidade e robustez em comparação ao controle efetuado normalmente. A execução do método proposto leva a realização de um controle otimizado sobre os índices de produção integrada, que explora uma nova abordagem mais segura, estável e econômica da operação do forno, com possibilidade de aplicação em vários processos industriais (AI-JUN, CHUN-XIAO, *et al.*, 2011).

O trabalho “*Adaptation methodology of CBR for environmental emergency preparedness system based on an improved genetic algorithm*”, propõe a criação de um sistema de raciocínio baseado em casos com algoritmo genético para resolver o problema de adaptação dos casos. Para as representações dos casos foram adotados frames. O algoritmo genético utilizado foi melhorado para garantir a eficiência da busca por soluções globalmente ideias, e foi chamado IGA (*Improved Genetic Algorithm*). O IGA emprega uma função de *fitness* multi-fator integrada, um método multi-parâmetro, e tecnologia *Niche* (LUO e GUIHUA, 2007) para operações genéticas, incluindo a seleção, cruzamento e mutação. Um protótipo CBR-IGA foi testado em um cenário de emergência ambiental. Foi introduzido, e sua aplicabilidade foi testada usando uma emergência hipotética de vazamento de amoníaco em um distrito de Shanghai. Os resultados indicam que a metodologia IGA proposta pode resolver o problema de adaptação e expandir a base de casos efetivamente em sistemas de raciocínio baseado em casos para preparação de emergência ambiental (LIAO, MAO, *et al.*, 2012).

O artigo “*An evolutionary approach to case adaptation*” apresentou um método de adaptação de casos utilizando algoritmo genético. Faz dois tipos de adaptação: combinação de casos e mutação de casos. Esses são usados para evoluir as variações dos casos sobre os casos recuperados até que seja encontrada uma solução satisfatória para o problema *alvo*. Variações sobre casos recuperados (indivíduos) são evoluídas de forma incremental, e em cada ciclo sua

qualidade é verificada, os melhores indivíduos compõem a população inicial, e são mantidos no ciclo atual (geração). A solução é considerada satisfatória se ela corresponde aos requisitos especificados e não viola quaisquer restrições impostas pelo domínio de aplicação. Esta solução computacional é chamada GENCAD e foi aplicada ao projeto de *layout* de residências com sua conformidade baseada nos princípios do *feng shui*, a arte chinesa de colocação. Esta aplicação permite-nos avaliar o uso do algoritmo genético para a adaptação do caso no sistema de raciocínio baseado em casos. Os resultados experimentais mostram a importância da representação e restrições (GARZA e MAHER, 1999).

No artigo “*Simple feasibility rules and differential evolution for constrained optimization*”, foi proposto um algoritmo de evolução diferencial para resolver problemas de otimização com restrições. Nessa abordagem utiliza três critérios de seleção baseados na viabilidade para orientar a busca para a região viável. A abordagem proposta não exige quaisquer outros parâmetros extras que normalmente são adotadas pelo algoritmo de evolução diferencial. Tal abordagem foi avaliada usando funções de teste a partir de uma referência conhecida comumente adotada para validar as técnicas de manipulação de restrições com algoritmos evolutivos. Estas funções estão descritas no trabalho de (RUNARSSON e YAO, 2000). Os resultados obtidos foram muito competitivos em relação as outras técnicas de manipulação de restrições que são representativas (MEZURA-MONTES, COELLO e TUN-MORALES, 2004).

O artigo “*Optimal VAR control for real power loss minimization using differential evolution algorithm*”, apresenta uma técnica de evolução diferencial com várias estratégias de geração de vetor. A metodologia proposta visa controlar as configurações de variáveis evitando perdas. O método foi examinado e testado, e os resultados obtidos são comparados com os outros dois métodos, nomeados, Método de Pontos (IPM), Nuvem de Partículas (PSO). O estudo comparativo mostra o potencial da abordagem proposta, sua eficácia e robustez para resolver problemas de engenharia (AMRANE, BOUDOUR, *et al.*, 2015).

No artigo “*Unsupervised feature selection using an improved version of differential evolution*”, um algoritmo de seleção de características não supervisionada foi proposto como uma versão melhorada de uma técnica de evolução diferencial. Ele produz um subconjunto de recurso ideal, otimizando três critérios: o desvio padrão médio do subconjunto de recursos selecionados, a dissimilaridade média dos recursos selecionados, e a semelhança média de recursos não-selecionados em relação aos seus primeiros recursos vizinho. Os resultados

experimentais confirmam a superioridade do algoritmo proposto sobre os outros algoritmos de seleção de características não supervisionada para oito tipos diferentes de conjuntos de dados (BHADRA e BANDYOPADHYAY, 2015).

No artigo “*A novel case adaptation method based on an improved integrated genetic algorithm for power grid wind disaster emergencies*”, foi proposto um método de adaptação de casos integrando um algoritmo genético multiobjetivo em um sistema de rede elétrica para prevenção de desastres provocados por ventos. O método proposto foi realizado em termos de relações correspondentes entre a similaridade do caso e plano de emergência, indicando uma nova ideia para a adaptação de casos. Um exemplo prático de aplicação foi testado no sul de *Jiangsu—China*— e mostrou que o método consegue um melhor desempenho de adaptação de casos, e também pode ser usado como um novo método para expandir a base do casos (ZHANG, LI e WANG, 2015).

O trabalho “*Case-base maintenance with multi-objective evolutionary algorithms*” diz que CBR é uma metodologia de resolução de problemas que utiliza a resolução de problemas antigos, os chamados casos, para resolver novos problemas. A base de casos é considerada uma fonte de conhecimentos. Nela os casos são armazenados e a quantidade de casos armazenados é fundamental para a capacidade de resolução de problemas do sistema CBR. No entanto, quando a base de casos tem muitos casos, surgem problemas de desempenho devido ao tempo necessário para encontrar esses casos semelhantes ao problema que se pretende resolver. Nesta situação, algoritmos de manutenção da base de casos podem ser utilizados para reduzir o número de casos e manter a precisão do sistema CBR. Esta situação pode ser considerada um problema de otimização multiobjetivo, e para resolver este problema foi utilizado um algoritmo evolutivo multiobjetivo com o NSGA-II. Para este fim, uma função de *fitness* foi medir três objetivos diferentes com base no modelo de perfil de complexidade. O objetivo foi que algoritmo evolutivo multiobjetivo realizasse a manutenção da base de casos, conseguindo um bom equilíbrio entre a redução do número de casos e a capacidade de resolução de problemas do sistema CBR. A partir de um conjunto de experimentos, o NSGA-II mostrou bons resultados com diferentes bases de casos, com diferentes proporções de casos, e com casos redundantes e ruidosos (LUPIANI, MASSIE, *et al.*, 2016).

O trabalho “*Utilização de algoritmo de evolução diferencial multiobjetivo no projeto de controladores*”, diz que o interesse do trabalho foi mostrar a utilidade dos algoritmos evolutivos, em especial o de evolução diferencial, no projeto automático de controladores.

Além disso, foi mostrada a utilidade da abordagem de otimização multiobjetivo para o problema de projeto de controladores. Com a abordagem multiobjetivo do problema, faz-se necessária a definição de um critério para a escolha de uma das soluções do conjunto de Pareto-Ótimo (conjunto de soluções viáveis). Uma possível definição é a escolha do controlador mais barato; neste trabalho foi escolhido o controlador cujo ponto na fronteira de Pareto seja mais próximo da solução ótima do problema. Desta maneira, basta que sejam definidas as restrições do problema, o modelo do projeto a ser controlado e a estrutura do controlador desejado e um critério para que um controlador ótimo seja definido automaticamente. Os resultados obtidos foram satisfatórios (SILVA, NETO e MARTINS, 2011).

O artigo “Um algoritmo baseado em evolução diferencial para problemas de otimização estrutural multiobjetivo com restrições”, relata que Problemas de otimização estrutural visam o aumento do desempenho da estrutura e a diminuição de seus custos garantindo, entretanto, os requisitos de segurança sejam mantidos. Devido à natureza conflitante desses aspectos, a formulação de um problema de otimização estrutural como multiobjetivo é natural, embora pouco frequente, e tem a vantagem de apresentar um conjunto diversificado de soluções ao(s) tomador(es) de decisão. A literatura mostra que os algoritmos evolucionários são eficazes na obtenção de soluções em problemas de otimização multiobjetivo e que aqueles baseados em evolução diferencial são eficientes na resolução de problemas de otimização estrutural monoobjetivo, especialmente os que utilizam codificação real em suas variáveis de projeto. Por outro lado, nota-se a ausência da aplicação da evolução diferencial na versão multiobjetivo desses problemas. Esse artigo apresentou uma análise do desempenho de um algoritmo baseado em evolução diferencial em cinco exemplos de problemas de otimização estrutural multiobjetivo. Os resultados obtidos são comparados aos encontrados na literatura, indicaram o potencial do algoritmo proposto (VARGAS, LEMONGE, *et al.*, 2016).

O artigo “*A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination*”, apresenta um estudo comparativo de diferentes abordagens de otimização meta-heurística para DOCRs – *Directional overcurrent relay coordination*). Para isto foram utilizadas cinco abordagens de otimização meta-heurísticas mais eficazes, como o algoritmo genético, otimização de enxame de partículas, evolução diferencial, busca de harmonia e algoritmo de otimização de busca. Os desempenhos desses métodos de otimização têm sido investigados em várias redes de sistemas de energia de diferentes tamanhos. Cada algoritmo foi executado 100 vezes com a mesma condição inicial, bem como com valores de parâmetro padrão. Com base nestes estudos, o algoritmo evolução diferencial

foi considerado o melhor entre os cinco algoritmos estudados, considerando: (a) o melhor valor obtido pelo evolução diferencial é sempre o mais baixo entre os melhores valores obtidos pelos outros algoritmos; (b) o melhor valor obtido pelo evolução diferencial é praticamente imune às variações dos parâmetros de entrada; (c) mesmo com diferentes curvas características dos relés, evolução diferencial obteve o menor valor da função objetivo. Com estes resultados, o artigo considerou o algoritmo evolução diferencial o mais adequado para a coordenação de DOCRs entre os algoritmos estudados (ALAM, BISWARUP e PANT, 2015).

Com relação aos trabalhos pesquisados foram relacionados alguns critérios utilizados pelos mesmos afim de nortear a escolha dos parâmetros de execução dos algoritmos utilizados nesta tese, conforme o Quadro 2.

Quadro 2. Visão consolidada dos trabalhos relacionados.

Referência	Método/Técnica/Algoritmo	População inicial	Mutação*	Crossover (%)	Gerações
(CHEN, LIU e CHOU, 2013)	NSGA-II MOP	100	3%	100%	60.000
(COELHO e ALOTTO, 2008)	NSGA-II MOP	30	2.5%	90%	200
(AI-JUN, CHUN-XIAO, <i>et al.</i> , 2011)	CBR MOP	N/A	N/A	N/A	N/A
(LIAO, MAO, <i>et al.</i> , 2012)	CBR AG MOP	20 a 100	10%	90%	100 a 500
(GARZA e MAHER, 1999)	CBR AG	12	#	#	500
(MEZURA-MONTES, COELLO e TUN-MORALES, 2004)	DE MOP	60	0.3 a 0.9	80% a 100%	5800
(AMRANE, BOUDOUR, <i>et al.</i> , 2015)	DE	150	0.5	60%	150
(BHADRA e BANDYOPADHYAY, 2015)	DE MOP	100	0.8	80%	100
(ZHANG, LI e WANG, 2015)	CBR AG MOP	#	#	#	50
(LUPIANI, MASSIE, <i>et al.</i> , 2016)	CBR NSGA-II MOP	30 a 50	3% a 5%	90% a 95%	100
(SILVA, NETO e MARTINS, 2011)	DE MOP	24	0.4	#	2400
(VARGAS, LEMONGE, <i>et al.</i> , 2016)	DE MOP	50	0.3	40%	1000
(ALAM, BISWARUP e PANT, 2015)	DE AG MOP	10 a 60	Randômica máximo de 0.2	Randômica máximo de 20%	100
<p>Legenda:</p> <p>CBR – Raciocínio baseado em casos</p> <p>DE – Evolução diferencial</p> <p>AG – Algoritmo genético</p> <p>MOP – Otimização multiobjetivo</p> <p>* – Para AG taxa de mutação em percentual (%), para DE constante de mutação (F).</p> <p># – Não especificado.</p> <p>N/A – Não se aplica.</p>					

Também foram pesquisados trabalhos relacionados ao domínio de aplicação, referentes a transporte ferroviário e condução automática de trens. No contexto de ferrovias, o artigo (GU, CAO e TANG, 2012) teve como objetivo determinar as velocidades praticadas durante a viagem do trem, utilizando programação não-linear com o intuito de evitar ações abruptas por parte do maquinista. Um foco semelhante foi dado por (HENGYU e HONGZE, 2012), porém foi utilizado redes neurais *fuzzy* para controlar as ações executadas na condução do trem. No trabalho (FANG, SUN, *et al.*, 2013), os autores apresentam diversas abordagens desenvolvidas para tratar o problema de reagendamento (alocação) de redes ferroviárias, e apresentam uma análise destas abordagens.

No contexto de condução de veículos, o trabalho (VACEK, GINDELE, *et al.*, 2007), propõe uma abordagem para condução autônoma de veículos. A abordagem utiliza CBR para com base na situação atual selecionar um comportamento apropriado de condução. Já o trabalho (LUO, JIYONG, *et al.*, 2014) propõe um modelo que faz uso de redes neurais para planejar a trajetória de um veículo inteligente, levando em consideração a segurança. Os resultados das simulações são apresentados para mostrar a eficácia e eficiência da metodologia proposta que realiza navegação livre de colisões e com segurança de um veículo inteligente em um ambiente dinâmico. O trabalho (MCDONOUGH, KOLMANOVSKY, *et al.*, 2011) utiliza *transition probability models* (TPMs) para modelar a condução de veículos para aplicações de otimização dinâmica. TPMs é utilizada para capturar padrões de estradas e de condução sem a necessidade de extração de características *ad hoc*; facilitando a aplicação dinâmica de um modelo de controle preditivo para melhorar a economia de combustível do veículo.

No contexto de transporte ferroviário, o trabalho (SABINO, LEAL, *et al.*, 2010) propõe um algoritmo de otimização com colônia de formigas. Objetivo é auxiliar a equipe de planejamento de operações de pátios ferroviários em suas tarefas diárias. Foi proposto um algoritmo baseado em uma função multiobjetivo, esta função buscou a redução dos custos fixo e variável de movimentação de vagões no pátio. Isto é feito através da busca da melhor programação para as locomotivas de manobra, considerando um dado horizonte de planejamento. O algoritmo foi desenvolvido para aplicação real, assim a solução deve ser fornecida em um tempo de processamento definido e obedecer às políticas operacionais do pátio.

No laboratório de pesquisa (LAS – Laboratório de Agentes de Software), quatro políticas para a condução trens foram implementadas, sendo elas: a primeira com aprendizagem

de máquina, a segunda com DCOP (*Distributed Constraint Optimization*), a terceira com especialização de regras (SATO, BORGES, *et al.*, 2012) e a quarta com CBR com 2 métodos de adaptação, sendo elas: substitutiva e algoritmo genético (BORGES, DORDAL, *et al.*, 2014). Considerando as pesquisas realizadas anteriormente, o presente trabalho, propôs um modelo capaz de recuperar e adaptar-se utilizando os casos mais similares para isso. Como resultado, foi esperado que o modelo além de conseguir aplicar eficientemente o ciclo de CBR também consiga garantir que a condução assistida seja mais rápida e econômica comparada a cenários reais de condução, utilizando para isso otimização multiobjetivo na adaptação de casos por meio do algoritmo genético e do algoritmo evolução diferencial.

2.11. Considerações do Capítulo

O ciclo de vida básico do raciocínio baseado em casos consiste em dado um problema-alvo recuperar casos similares. Esta etapa deve ser eficiente, pois deve validar com eficácia o quanto um caso é semelhante ao problema-alvo. Após o processo de recuperação, o(s) caso(s) similar(es) passa(m) por um processo de adaptação. Neste processo de adaptação diversos métodos podem ser empregados. Basicamente, tais métodos são classificados em: métodos que exigem pouco conhecimento (estocásticos e determinísticos) e métodos que exigem grande volume de conhecimento (substituição, transformação e *ranking* de casos recuperados). O caso adaptado é testado afim de validar sua eficácia. O caso aprovado no teste é aplicado e armazenado. O caso reprovado é reparado até que seja aprovado no teste e possa ser aplicado. Para construir este ciclo as mais diferentes técnicas e algoritmos podem ser empregados nas etapas que o compõem.

Quanto otimização multiobjetivo e seus métodos, o desafio foi mapear e discutir a produção acadêmica acerca destes tópicos principais e seus subtópicos, tentando apresentar conceitos e principalmente aplicações relevantes dos mesmos. Utilizando este levantamento para nortear esta tese.

Os métodos de otimização multiobjetivo podem ser classificados em: métodos *a priori*, métodos *a posteriori* e métodos automáticos. Os métodos com articulação *a priori* permitem ao usuário especificar suas preferências como restrições e/ou objetivos *vis-à-vis* o problema-alvo. Os métodos com articulação *a posteriori* permitem ao usuário efetuar escolhas dado um conjunto de soluções. E os métodos sem articulação de preferências são métodos que não

exigem que o usuário especifique suas preferências. Juntamente com o método adotado (*a priori*, *a posteriori* ou automáticos), pode-se empregar para solucionar problemas de otimização multiobjetivo diversos algoritmos evolutivos; como algoritmo genético, evolução diferencial, nuvem de partículas, entre outros.

Assim para esta tese foi adotado o método de otimização multiobjetivo com articulação *a posteriori* das preferências e combiná-lo com algoritmos de base populacional, sendo eles o algoritmo genético e o algoritmo evolução diferencial. Estas escolhas foram feitas considerando as pesquisas realizadas e os trabalhos relacionados.

Capítulo 3

Método Proposto

O presente trabalho propõe um método eficiente para a geração de planos de condução de trens interurbanos de carga. Parte da eficiência do método decorre da abordagem básica adotada para reuso de experiências, cujo principal esforço encerra-se em torno do problema de adaptação de casos—ou de adaptação de planos. Cada plano é formado por um conjunto ordenado de sub-planos. Cada sub-plano reúne as ações que devem ser tomadas para vencer um perfil dominante de um dado sub-trecho de uma via férrea, por exemplo, aclive, declive, em nível, entre outros. Um gerador de planos neste contexto foi inicialmente proposto em (BORGES, 2015). A especificidade desta tese é gerar sub-planos utilizando o conceito de otimização multiobjetivo aliado a abordagem de adaptação de casos. Cada sub-plano pode ser visto como um caso.

O quadro geral da aplicação foi proposto em (BORGES, 2015). Assim, o leitor está convidado a ler a referida referência caso o seu interesse seja também obter uma visão sobre o modal férreo, em particular, a geração de políticas de condução para trens interurbanos de carga distribuída sobre uma arquitetura de agentes autônomos.

Para embasar a tomada de decisão e as escolhas feitas foram testadas duas abordagens de adaptação, sendo elas abordagem genética (algoritmo genético) e abordagem diferencial evolutiva (algoritmo de evolução diferencial).

3.1. Domínio de Aplicação

A condução de trens interurbanos de carga é uma tarefa complexa, à medida que o conjunto de conhecimentos e habilidades necessárias para uma condução segura e eficiente é diferente daqueles observados na condução de veículos de passeio e, em certos trechos, somente condutores com uma grande experiência podem viajar. Por exemplo, um fator que dificulta a condução de trens de carga é a aplicação de freios. Em veículos de passeio, a aplicação de freios é praticamente instantânea, já em grandes (uma centena de vagões) trens de carga a aplicação de freios pode levar alguns minutos, além de depender de fatores como: tamanho do trem, da posição da alavanca de configuração de freios (a qual delimita o tempo de aplicação de ar no sistema), modelo de sistema de freios disponíveis nos vagões, entre outros (BORGES, 2015), (DORDAL, 2015)

A seguir será apresentada de forma resumida a arquitetura do simulador de condução de trens interurbanos de carga desenvolvido no laboratório de Agentes de Software do Programa de Pós-Graduação da PUCPR.

3.2. Ambiente de Condução

Esta seção foi baseada na tese (BORGES, 2015) e na tese (DORDAL, 2015), nas quais o ambiente de condução foi especificado em detalhes, e foram utilizados como base para este trabalho. A malha viária é formada por um conjunto de estações conectadas por vias férreas sob as quais trafegam um ou mais trens. O sistema de condução fornece um conjunto bem-definido de serviços, a saber: executar ordens de viagens expedidas pelo sistema gerenciamento de malha férrea, criar e executar expectativas de condução de trens para locomotivas *monoponto*, gerenciar experiências obtidas por meio da adaptação de casos no decorrer de uma viagem para uma posterior reutilização. Cada uma destes serviços é operacionalizado por um agente de software. Esses agentes são:

- *Agente Despachante*. Localizado na central de monitoramento e controle, tal agente tem como função informar as licenças de movimentação na malha férrea, ou seja, autorizar a partida e parada de trens.
- *Agente Planejador*. Localizado em cada estação de trem, tal agente tem como função adaptar, validar, reparar (se necessário) planos de condução.

- *Agente Executor*. Localizado no computador de bordo da locomotiva principal, ele tem a função de reparar as ações de um plano (se necessário) e aplica-las.
- *Agente Memorizador*. Localizado em cada estação de trem, ele tem a função de recuperar casos passados e reter novos casos/planos.

Estes agentes interagem promovendo assim a execução do ciclo canônico de raciocínio baseado em casos, ora resumido a seguir e detalhado mais a frente: o agente despachante envia ao agente planejador uma demanda (ordem de despacho) para rebocar um trem do ponto A até o ponto B (problema-alvo). Neste momento, o agente planejador segmenta o percurso de A até B em sub-trechos. Com as informações de cada sub-trecho, o planejador envia uma requisição de planos similares ao agente memorizador; cada requisição encerra um sub-trecho e dá forma ao enunciado do problema-alvo. O agente memorizador retorna os casos mais similares (a medida de similaridade empregada foi a distância Euclidiana). Estes casos passam pelo processo de adaptação, resultando em um conjunto de soluções viáveis, das quais uma delas é escolhida e passa a ser a solução do problema-alvo e posteriormente retido pelo agente memorizador. O plano—formado por um conjunto ordenado de sub-planos/casos adaptados—é executado pelo agente executor. Deve-se notar que se houver a necessidade de reparo de uma ação de um plano, ela será efetuada, e após tal reparo a mesma é aplicada; o reparador aqui é o agente executor e essa tarefa é feita em tempo real.

Assumimos que uma malha férrea é um grafo. Lembramos que um grafo é um par $G = (V, E)$, onde V é um conjunto de n vértices $\{v_1, v_2, \dots, v_n\}$ e E é um conjunto de n arestas $\{e_1, e_2, \dots, e_n\}$; cada aresta é formada por um par de vértices $v = (u, v)$ em que v e $u \in V$. Mapeando então, uma malha férrea para um grafo, V é um conjunto de estações férreas, i.e., $V = \{s \mid s \text{ é uma estação}\}$, e E é um conjunto de caminhos que conectam as estações, logo, $E = \{(u, v) \mid \langle u \text{ está ligada à } v \rangle\}$. Exemplificando tal situação, sendo $V = \{A, B, C\}$ as estações férreas e $E = \{(A, B), (A, C), (B, A), (B, C), (C, A), (C, B)\}$ os caminhos entre as estações de V . A relação $\langle u \text{ está ligada à } v \rangle$ é simétrica, uma vez que se $\langle u \text{ está ligada à } v \rangle$ então $\langle v \text{ está ligada à } u \rangle$.

O grafo G que representa uma malha férrea é rotulado (cf. Figura 4), à medida que cada vértice possui um rótulo associado. Cada rótulo identifica um conjunto de dados a respeito do trecho de via férrea que ele representa, denominado conjunto, $PM = \{pm \mid pm \text{ é um ponto de medida}\}$.

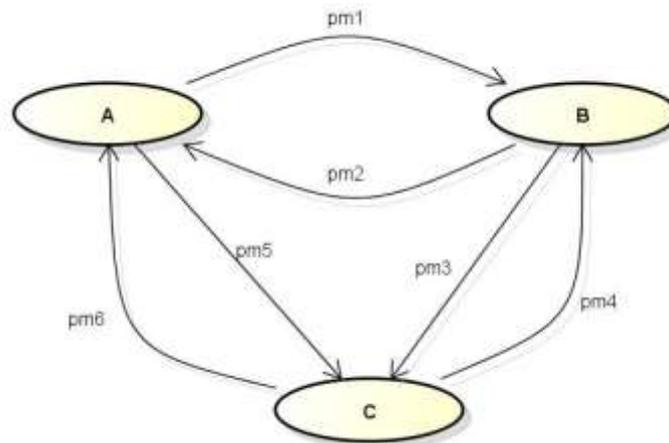


Figura 4. Exemplo de malha férrea representada por um grafo rotulado. Cada aresta representa um conjunto de pontos de medida.

O grafo G que representa uma malha férrea além de ser rotulado cf. Figura 5, é orientado (i.e., dígrafo) (ROSEN, 2012). Os vértices são conectados por arestas orientadas. Isto permite representar o sentido do caminho percorrido por um trem na malha férrea. Um dígrafo é um par $G_u = (V, E)$, onde V é o conjunto de vértices e E é o conjunto de arestas. Cada elemento de E é par ordenado: a aresta do vértice u ao vértice v é escrita como (u, v) e o par (v, u) é a aresta na direção oposta. Um caminho em dígrafo pode ser escrito como: $v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \dots, e_{j_k}, v_{i_k}$, onde v_{i_l} é o vértice inicial e $v_{i_{l-1}}$ é o vértice final da aresta e_{j_l} . Logo, $G = G_u$.

O grafo G que representa uma malha férrea além de ser um dígrafo rotulado, é um multigrafo (ROSEN, 2012). Além do sentido entre estações de uma malha férrea pode haver dois caminhos diferentes que conectam duas estações. Um multigrafo orientado permite que múltiplas arestas m conectem pares de vértices de G , cada aresta possui uma orientação. Formalmente, um multigrafo orientado é uma tripla $G = (V, E, f)$, onde V é o conjunto de vértices, E é o conjunto de arestas e $f: E \rightarrow V \times V$ é uma função (ROSEN, 2012).

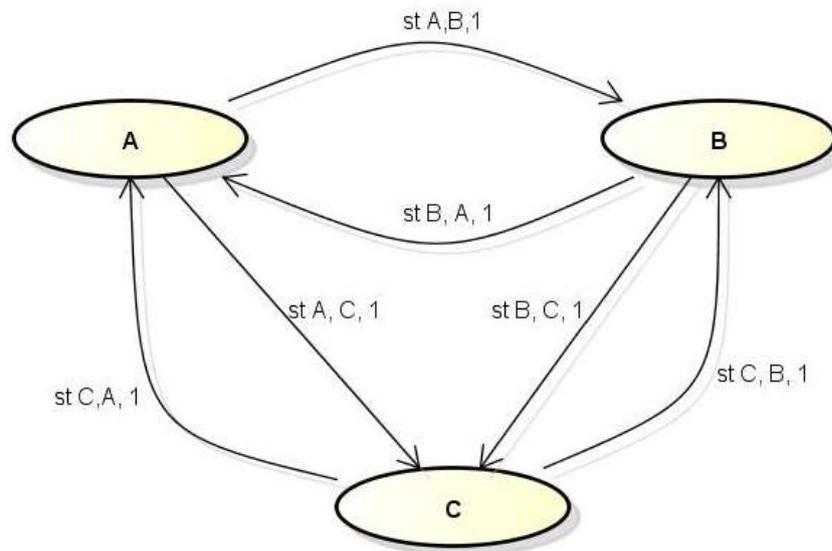


Figura 5. Exemplo de malha férrea representada por um grafo rotulado com caminhos entre as estações.

O conjunto de vértices V de G corresponde às estações férreas S , logo $V = S$, onde S é um conjunto de n estações s_{id} , e cada estação é identificada univocamente por índice id . Assim sendo, $S = \{s_{id} \mid id = 1, 2, \dots, n\}$, onde $n = |V| = |S|$.

O conjunto de arestas E de G corresponde aos caminhos trafegáveis por trens, de estação em estação. A orientação de cada aresta indica o sentido trafegável, que parte do vértice v_i para o vértice v_j . Como G é um multigrafo, o tráfego pode ocorrer por qualquer caminho k dado por $f: E \rightarrow V \times V$. Para expressar tal relação, fez-se uso da notação $E_{i,j,k}$, onde E é igual à um trecho/caminho de via férrea St , i a estação de origem, j a estação de destino e k o caminho a ser percorrido, assim temos que $E_{i,j,k} = St_{i,j,k}$.

Cada trecho físico de via férrea $St_{i,j,k}$ é descrito por um conjunto ordenado de pontos $p \in PM_k$, obtidos do projeto topográfico da ferrovia considerada (Projeto Planimétrico) e normalmente calculados de 20m em 20m (CHANDRA e AGARWAL, 2007). Cada ponto pm —ou ponto de medida— é um registro estruturado de dados estrutura contendo: id (identificador do ponto de medida), km (Km de referencia), velMax (velocidade máxima permitida em Km/h), rampa (inclinação da rampa em percentual), raioCurva (raio da curva em metros), AC (ângulo central da curva em metros), g20 (grau da curva para uma corda de 20 metros), altitude (altitude em metros), nroSb (identificador da seção de bloqueio a qual o ponto de medida pertence). A Tabela 1 fornece uma descrição para cada componente deste registro e ilustra um trecho $St_{A,B,1}$ que conecta as estações A e B pelo caminho 1, cuja extensão é de 140 metros.

Tabela 1. Exemplo de descrição de sub-trecho de via férrea com sete pontos de medidas, extraídos do projeto topográfico de uma via férrea.

ID	KM	VELMAX	RAMPA	RAIOCURVA	AC	G20	ALTITUDE	NROSB
1	204	60	0.95	407.5	35.96	2.81	526.87	3
2	204	60	0.95	407.5	35.96	2.81	526.87	3
3	204	60	0.95	407.5	35.96	2.81	526.87	4
4	204	60	0.95	407.5	35.96	2.81	526.87	4
5	205	60	0.95	407.5	35.96	2.81	526.87	4
6	205	60	1.20	818.52	66.20	1.40	534.26	4
7	205	60	1.20	818.52	66.20	1.40	534.26	4

Um caminho k que liga duas estações s_i e s_j , é designado por um sub-trecho de via férrea $ST_{s_i,s_j,k}$, onde s_i é a estação de origem, s_j é a estação de destino; lembramos que entre s_i e s_j podem haver mais de um caminho k . Em termos de notação, lembramos que o conjunto de vértices V de G corresponde às estações férreas S , logo $V = S$, e S é um conjunto de n estações s_{id} , onde cada estação é identificada univocamente por índice id . Assim sendo, $S = \{s_{id} \mid id = 1, 2, \dots, n\}$, onde $n = |V| = |S|$. Assim, ST_k será utilizado para designar um trecho de via férrea $ST_{i,j,k}$.

A Figura 6 mostra um trecho de via férrea $ST = \{st_1, st_2, \dots, st_n\}$, onde cada st_i é um segmento cujos limites st_i de início e fim são determinados pelo perfil, podendo ser: aclave, declive ou em nível. Cada st_i é tratado de forma individual. I.e., para cada situação pr_i — definida sobre o conjunto de pontos de medidas do segmento de via férrea $st_i(ID_i, KM_i, VelMax_i, Rampa_i, RaioCurva_i, AC_i, G20_i, Altitude_i, NroSb_i)$ —, um caso c_i é recuperado da base de casos CB e um problema-alvo $pb_i = \langle pr_i, c_i \rangle$ é instanciado. A solução de cada pb_i é dada por meio da adaptação do conjunto de pontos de aceleração de $ap_j \in c_i$, nesta adaptação foram utilizados duas abordagens evolucionistas: genética e diferencial evolutiva; ambas operando em um cenário multiobjetivo. Feita a adaptação, uma solução é escolhida. Tal solução é testada. Este teste consiste em aplicar os pontos de aceleração $j \in AP'$ e armazenar pb_i em CB_{local} ; obviamente se o plano adaptado for reprovado no teste, ele é reparado no final deste processo obtém-se $P = \{\langle st_1, p_1 \rangle, \langle st_2, p_2 \rangle, \dots, \langle st_n, p_n \rangle\}$, que é o plano completo para rebocar um trem T , sobre o trecho ST , usando o plano P . Cada p_i descreve um conjunto de ações com os comportamentos (e.g., manter, acelerar, reduzir ou frear) para um perfil da via (e.g., aclave, declive e em nível).

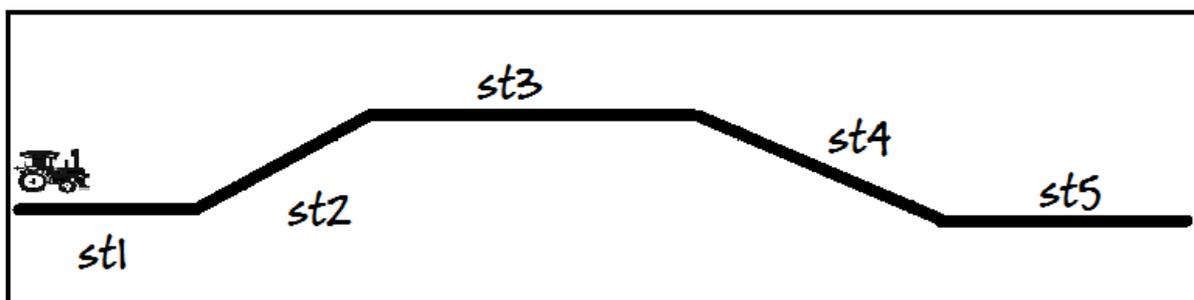


Figura 6. Linha férrea ST, dividida em trechos st_n .

O perfil é determinado de acordo com o percentual de rampa. A classificação do perfil foi realizada com base nos estudos publicados em (PERREIRA, 1958). Esses estudos são ainda utilizados pelo Departamento Nacional de Infraestrutura de Transportes (DNIT). Segundo Perreira, para trens com bitola de 1.60m, um trecho plano possui um percentual de rampa (%R) máxima de 0.06. Portanto, trechos planos são aqueles cujo percentual de rampa (%R) está $[-0.06, 0.06]$. Trechos de aclive são aqueles com $\%R > 0.06$ e trechos em declive aqueles com $\%R < -0.06$.

O módulo de raciocínio baseado em casos utiliza em sua base de experiências passadas casos cuja representação segue o padrão atributo-valor, sendo utilizado os seguintes atributos: ação aplicada (*EAC*), tipo do perfil a percorrer (Perfil), quilômetro inicial (*KM*), número de locomotivas (*NL*), número de vagões (*NC*), velocidade inicial (*IS*) em km/h, velocidade final (*FS*) em km/h, velocidade máxima (*MS*) em km/h, percentual de rampa (%R), deslocamento total (*CL*) em quilômetros e o sub-planos (*SP*). O Quadro 3 mostra a representação de um caso.

Quadro 3. Representação de um caso.

Atributo	Valor	Significado
<i>EAC</i>	Acelerar	Ação aplicada (Acelerar, Manter, Frenar)
<i>Perfil</i>	Inclinado	Perfil da via
<i>KM</i>	339	Quilometro inicial em Km
<i>NL</i>	3	Número de locomotivas
<i>NC</i>	58	Número de Vagões
<i>IS</i>	10	Velocidade inicial em Km/h
<i>FS</i>	11	Velocidade final em Km/h
<i>MS</i>	45	Velocidade máxima em Km/h
<i>%R</i>	0.41	Percentual de rampa
<i>CL</i>	0.403	Deslocamento total em Km
<i>SP</i>	{< 2,0 >, < 2,5 >, < 4,24 >}	Sub-planos

Os casos organizados de acordo com as ações tomadas e os meta dados da via férrea (perfil, km, % de rampa), deslocamento do trem coberto pelo plano/caso (CL), e os sub-planos (SP), que é composto por pares de valores, que indicam respectivamente o ponto de aceleração aplicado ap_i e posição de sua aplicação (m_i). Esta organização permite recuperar os casos mais similares ao trecho da via férrea em questão e a ação que foi aplicada (cf. Tabela 2).

Tabela 2. Representação da base de casos.

CASO	AÇÃO	PERFIL	KM	%R	CL	m_1	ap_1	m_2	ap_2	m_i	ap_i
...
102	Acelerar	Em Nível	242424	-0.90	0.503	0	4	35	5		
103	Acelerar	Em Nível	565433	-0.04	0.900	0	3	15	4
104	Manter	Em Nível	452357	-0.04	0.456	0	3	45	3		
...

Para a recuperação de casos na base de casos é calculada a similaridade entre o problema-alvo e os casos registrados. Este cálculo é feito por meio da distância Euclidiana. Assim, os casos mais semelhantes ao problema-alvo são retornados.

3.3. Método

A visão geral do método está centrada na abordagem clássica de um solucionador de problemas baseados em casos. A Figura 7 dá a estrutura de tal solucionador de problemas (visto em detalhes na seção 2.2). Os esforços estão centrados na etapa de adaptação de casos. A especificidade está na otimização multiobjetivo de cada caso. Para tal são usados duas abordagens de otimização evolucionistas: abordagem genética multiobjetivo e a abordagem diferencial evolutiva multiobjetivo. As partes da Figura 7 na cor vermelha destacam as contribuições, considerando que uma adequada adaptação de casos tem influência em outras etapas do ciclo (recuperação e teste), uma vez que a adaptação produza soluções com qualidade, estas são armazenadas na base de casos e reaproveitadas futuramente.

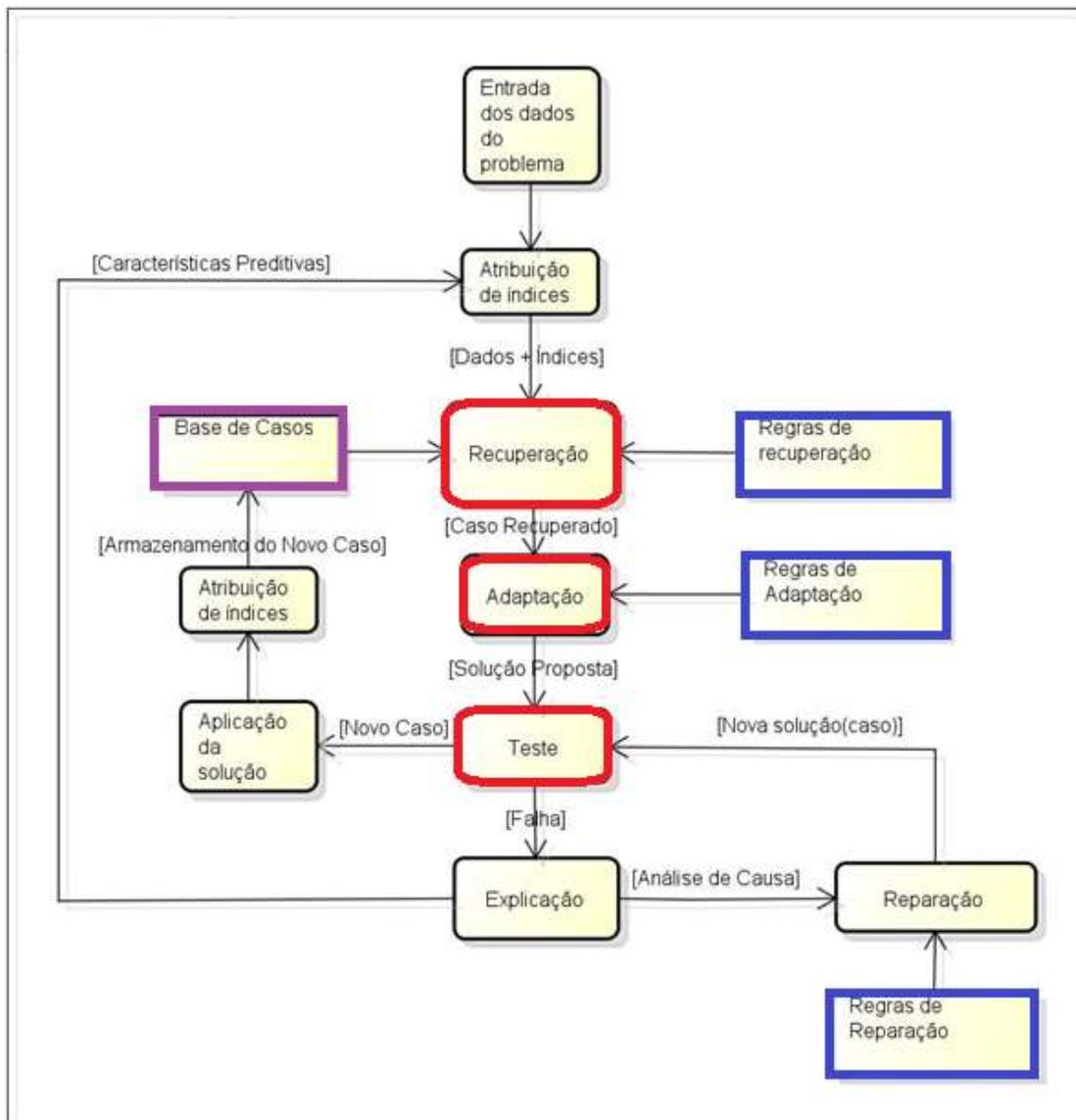


Figura 7. Ciclo clássico CBR destacando contribuições do trabalho.

Para a recuperação de casos é calculada a similaridade entre o problema-alvo e os casos registrados na base de casos. Este cálculo é feito usando a distância *Euclidiana*. Assim, os casos mais semelhantes ao problema-alvo são retornados. Os casos retornados dão origem a população inicial $P_{inicial}$, e esta é utilizada pelo algoritmo de adaptação (algoritmo genético ou evolução diferencial).

3.3.1. Funções Objetivo

Nesta tese optou-se por usar a abordagem¹ multiobjetivo, assim foram definidas 2 funções objetivo, sendo elas: minimizar o consumo de combustível $LTKB$ (f_1) e maximizar a velocidade do trem v (f_2). São atribuídas penalidades w à cada função objetivo. Penalidades são atribuídas em situações que ocasionem falta de segurança, danos a via férrea ou ao trem. Se algum indivíduo gerado se enquadra em alguma destas condições, o seu valor de fitness é igual a $100/(tamanhoGene/2)$. O valor 100 é significativo para os valores médios obtidos pelas funções objetivo, e foi definida considerando os trabalhos (BORGES, 2015) (LEITE, 2009).

O Quadro 4 resume as condições que geram penalizações.

Quadro 4. Condições de penalidade.

CONDIÇÃO	DESCRIÇÃO
$v \geq v_{max}$	Risco eminente de acidente: velocidade corrente (v) superior a velocidade máxima (v_{max}).
$F_t > (F_{tm} \times nl)$	Risco de patinagem: esforço trator (F_t) superior ao produto entre esforço trator máximo (F_{tm}) e o número de locomotivas tracionando (nl).
$\Delta\ell \geq \Delta\ell_{max}$	Risco de descontrole: deslocamento da ação ($\Delta\ell$) é maior que o deslocamento máximo previsto ($\Delta\ell_{max}$).
$F_{ac} < 0$ e $\Delta v > 0$	Risco de descontrole: aumento de velocidade ($\Delta v > 0$) sem força de aceleração suficiente ($F_{ac} < 0$).
$F_{ac} > 0$ e $\Delta v < 0$	Risco de parada: redução de velocidade ($\Delta v < 0$) com força de aceleração positiva ($F_{ac} > 0$).
$v_F > v_{max}$ e $v_F > v$	Risco eminente de acidente: velocidade final (v_F) superior a velocidade máxima e encontra-se em aceleração ($v_F > v$).

¹ Esta abordagem foi escolhida porque uma abordagem utilizando soma ponderada como função de *fitness* já havia sido utilizada no trabalho de (BORGES, 2015) e havia uma hipótese de que a abordagem multiobjetivo poderia apresentar resultados mais satisfatórios.

3.3.2. Representação dos Indivíduos

Foram determinadas características diferentes para os *cromossomos* pares e ímpares dos indivíduos. Assim, os genes pares representam pontos de aceleração (ap_i) da locomotiva líder e ímpares representam as posições de aplicação dos pontos de aceleração sobre um dado trecho de uma via férrea. O valor do alelo corresponde ao local (m_j) de aplicação (em metros) de um ponto de aceleração (ap_i) a partir do primeiro gene, de modo que o alelo do primeiro gene ímpar é sempre igual a zero, pois ele representa o ponto inicial definido para o caso a ser adaptado (Figura 8). Quanto ao número de *cromossomos*, optou-se por um tamanho fixo, com 5 pares (10 genes).

ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
5	0	3	5	4	25	6	38	8	50

Figura 8. Representação de um indivíduo.

3.3.3. Mutação

A operação de mutação troca valores de alguns genes da população. A mutação feita no contexto de um algoritmo genético e de um algoritmo de evolução diferencial são diferentes.

O processo de escolha de quais genes sofrerão mutação foi definida por uma função de geração de valores aleatórios fa , em que P_n é a população atual. Caso um dado gene seja escolhido, ele receberá um valor aleatório, gerado pela função $fa(i,j)$ que recebe como parâmetros um valor mínimo (i) e um valor máximo (j).

$$fa(i,j) = \{Rand(P_n)\} \quad (15)$$

Após a mutação, os melhores indivíduos são selecionados. Para a mutação utilizou-se uma probabilidade de 5% (cinco por cento). Sendo assim, a mutação para a abordagem genética foi realizada cf. Figura 9.

Gene escolhido	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		5	0	3	5	4	25	6	38	8

Gene após mutação	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		5	0	3	7	4	25	5	37	7

Figura 9. Exemplo de mutação para a abordagem algoritmo genético.

A mutação para a abordagem diferencial evolutiva foi realizada da seguinte maneira: são escolhidos 2 genes de forma aleatória, a partir deles foi gerado um vetor de diferença, que é a diferença entre o gene 1 e o gene 2. Um gene 3 é escolhido também de forma aleatória. Este gene é somado ao vetor de diferença e o resultado é chamado de solução mutante (cf. Figura 10).

Gene 1 Escolhido	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		5	0	3	5	4	25	6	38	8

Gene 2 Escolhido	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		3	0	4	6	4	23	5	35	7

Vetor de Diferenças x F	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		2	0	-1	-1	0	2	1	3	1

Gene 3 Escolhido	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		4	0	4	6	5	22	5	37	7

Solução Mutante	ap_1	m_1	ap_2	m_2	ap_3	m_3	ap_4	m_4	ap_5	m_5
		6	0	3	5	5	24	6	40	8

Figura 10. Exemplo de mutação para a abordagem evolução diferencial.

A solução mutante ainda é multiplicada por F . F é a constante de mutação ou fator de mutação. O valor arbitrado foi de 1. Em (STORN e PRICE, 1995) sugere arbitrar um valor entre 0 e 2. O valor 1 não deve influenciar os resultados, na medida em que o valor de F determina o tamanho do passo a ser dado na direção definida pelo vetor diferença. Caso a solução mutante gerada extrapole algum valor permitido, por exemplo ap_n varia de -1 até 8, e foi gerado um valor 9, este valor será reduzido até o valor mais próximo permitido, i.e., neste caso 8.

3.3.4. Cruzamento

O cruzamento possibilita a recombinação das estruturas genéticas de uma população, permitindo diversificar o espaço de configurações por gerar indivíduos diferentes dos atuais. A combinação de cromossomos é feita de dois em dois indivíduos de uma população.

Foi adotado o *cruzamento* de 1-ponto tanto para a abordagem genética quanto para a abordagem diferencial evolutiva, na medida em que esta estratégia obteve os melhores resultados quando comparada a estratégias de cruzamentos em problemas clássicos de otimização combinatória (MENDES, 2013) (PICEK, GOLUB e JAKOBOVIC, 2012).

O cruzamento é aplicado da seguinte forma (cf. Figura 11): dados 2 pais (*caso* c_1 e *caso* c_2), uma posição p (ponto de corte) é determinada, o primeiro filho (*novo* c_1) recebe os genes do primeiro pai (*caso* c_1) até a posição p , e os genes do segundo pai (*caso* c_2) a partir da posição p . O segundo filho (*novo* c_2) recebe os genes do segundo pai até o ponto p e os genes do primeiro pai de p em diante. O ponto de corte é determinado aleatoriamente.

<i>caso</i> ₁	4	0	2	20	5	15	6	85	7	70
<i>caso</i> ₂	5	0	4	23	5	33	10	77	8	90
<i>novo caso</i> ₁	4	0	2	20	5	33	10	77	8	90
<i>novo caso</i> ₂	5	0	4	23	5	15	6	85	7	70

Figura 11. Cruzamento de 1-ponto.

3.3.5. Seleção

A operação de seleção é realizada para cada geração. Ela determina quais indivíduos da população atual passarão para a próxima geração. Neste trabalho, a seleção foi feita usando o conceito de dominância para determinar a estratégia de classificação e assim efetuar a seleção/classificação dos indivíduos. Para tal foram utilizados no processo de seleção dois mecanismos: *Fast Non-Dominated Sorting* (cf. 2.9.1) e *Crowding Distance* (cf. 2.9.2).

3.3.6. Adaptação dos Casos

Para adaptar um caso, primeiramente, é feita uma identificação do que é necessário alterar. Isto é realizado por meio de uma lista de verificações. Esta lista é formada por um conjunto de normas que garantem a movimentação de um dado trem, como: restrição de velocidade máxima, patinagem, falta de força e consumo excessivo. Recuperado um caso, o primeiro passo é verificar se cada ação sugerida é passível de aplicação seguindo a lista de verificação. Esta verificação permite modificar o caso para eliminar quaisquer umas das violações de normas/restrições existentes.

O processo de adaptação é executado por 10 gerações (critério de parada) e encerra fornecendo uma população de indivíduos (soluções viáveis). Como foi adotado um método de escolha *a posteriori*. O indivíduo selecionado é aquele que resulta no menor consumo de combustível e na maior velocidade. Este indivíduo é aquele que possui o menor valor de f_1 e o maior valor de f_2 . Ele é considerado uma solução viável para *problema-alvo*.

Neste processo de adaptação (cf. Figura 12), foi usado uma abordagem genética (algoritmo genético multiobjetivo) e uma abordagem diferencial evolutiva (algoritmo evolução diferencial multiobjetivo); os resultados obtidos são relatados nos próximos capítulos.

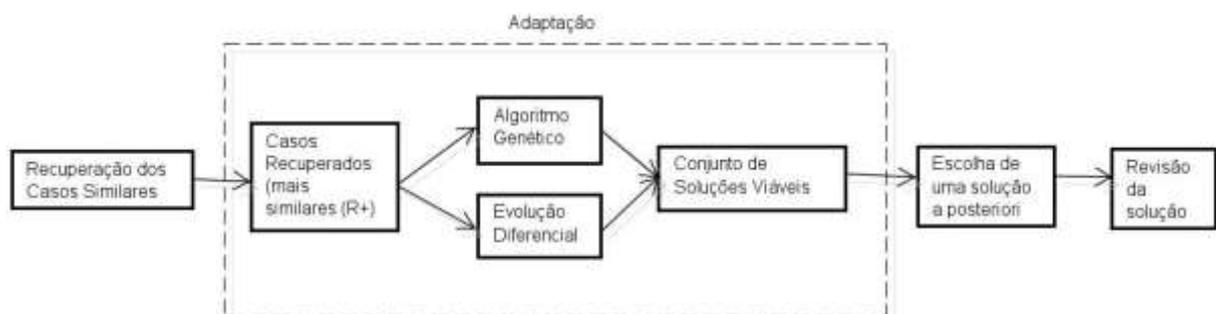


Figura 12. Processo de adaptação.

3.3.7. Revisão

O caso que descreve a solução U (um sub-plano) para o *problema-alvo*, deve ser revisado antes dele ser adicionado ao plano de condução. Este procedimento consiste em avaliar a

aplicabilidade de cada ação a_k do par da solução U . Os critérios de validação são apresentados no Quadro 5.

Quadro 5. Critérios de validação de uma solução.

Fonte: (BORGES, 2015).

FUNÇÃO	DESCRIÇÃO
$validar(a_k) = \begin{cases} SUCESSO & \text{if}(testar(a_k) = 0) \\ explicar(testar(a_k)) & \text{c. c.} \end{cases}$	<p>Uma explicação é gerada se a solução proposta não for considerada aplicável.</p>
$testar(a_k) = força(a_k) + patinagem(a_k) + consumoExcessivo(a_k)$	<p>A soma igual à zero indica sucesso.</p>
$patinagem(a_k) = \begin{cases} 1 & \text{if } F_t > (F_{tm} \times nl) \\ 0 & \text{c. c.} \end{cases}$	<p>O valor 1 indica patinagem, F_t é o esforço trator e F_{tm} é o esforço trator máximo, nl é o número de locomotivas tracionando.</p>
$força(a_k) = \begin{cases} 3 & \text{if } F_{ac} \leq 0 \\ 0 & \text{c. c.} \end{cases}$	<p>O valor 3 indica falta de força de movimentação, onde F_{ac} é a força de aceleração da ação.</p>
$consumoExcessivo(a_k) = \begin{cases} 7 & \text{if } F_{ac}(ap_{i-1}) > 0 \\ 0 & \text{c. c.} \end{cases}$	<p>O valor 7 indica excesso de consumo de combustível, onde $F_{ac}(ap_{i-1})$ é a força de aceleração empregada por uma ação com uma unidade inferior ao ciclo atual.</p>
$explicar(x) = \begin{cases} \{patinagem\} & \text{if } x = 1 \\ \{faltaForça\} & \text{if } x = 3 \\ \{consumoExcessivo\} & \text{if } x = 7 \\ \{patinagem, faltaForça\} & \text{if } x = 4 \\ \{patinagem, consumoExcessivo\} & \text{if } x = 8 \\ \{faltaForça, consumoExcessivo\} & \text{if } x = 10 \\ \text{todas as condições} & \text{if } x = 11 \end{cases}$	<p>Indica o tipo de falha, onde <i>patinagem</i> indica condição de patinagem, <i>faltaForça</i> indica falta de força para rebocar o trem e <i>consumoExcessivo</i> indica excesso de consumo de combustível.</p>
$reparar(x) = \begin{cases} 1 - & \text{if}(X \subseteq \{patinagem, consumoExcessivo\}) \\ 1 + & \text{if}(X \subseteq \{faltaForça\}) \\ - & \text{caso contrário} \end{cases}$	<p>Reparação da potência selecionada para uma ação; indica se o ponto de aceleração deve ser decrementado (1-) ou incrementado (1+) de uma unidade.</p>

3.3.8. *Método de Avaliação*

Para testar a eficiência da abordagem proposta, diferentes experimentos computacionais foram realizados. As métricas utilizadas para avaliar o desempenho foram as seguintes: consumo de combustível, tempo de viagem e taxa de acerto (no processo de recuperação dos casos e no processo de adaptação dos casos). Esta avaliação foi feita com dois propósitos:

- Primeiro propósito: comparar os resultados obtidos das abordagens evolucionistas: algoritmo genético e evolução diferencial. Tais abordagens devem operar em um cenário de otimização multiobjetivo—tempo de viagem e consumo de combustível.
- Segundo propósito: comparar a taxa de acerto da tarefa de recuperação de casos, e da tarefa de adaptação de casos usando abordagens evolucionistas: algoritmo genético e evolução diferencial.

Deve-se salientar que a consecução de ambos propósitos foi feita considerando diferentes cenários realistas de condução de trens interurbanos de carga, bem como diferentes configurações de trens e de vias férreas.

3.4. **Considerações do Capítulo**

Este capítulo apresentou o método desenvolvido no presente trabalho de pesquisa. O método desenvolvido foi centrado na etapa de adaptação de casos do ciclo canônico de CBR. Optou-se por centrar os esforços na etapa de adaptação de casos, à medida que a eficiente tarefa de adaptação de casos propicia uma eficácia geral, resultando em uma base de casos consistente, significativa e representativa. A qualidade dos casos tem impacto direto na eficiência da tarefa de recuperação de casos—pode ser mais assertiva—, assim como na tarefa de adaptação de casos.

Considerando o foco ser na etapa de adaptação de casos e já ter conhecimento que o trabalho de (BORGES, 2015) obteve resultados promissores utilizando a abordagem genética com soma ponderada, optou-se pela abordagem genética multiobjetivo. Também foi considerado que a abordagem genética multiobjetivo é bastante difundida na literatura. Com o objetivo de comparar os resultados da abordagem genética com outra abordagem, optou-se pela

abordagem evolutiva diferencial; ela tem desempenho significativo vis-à-vis métodos evolucionistas multiobjetivo. Finalmente, ambas abordagens—genética e diferencial evolutiva—atendem uma característica do problema de pesquisa tratado, i.e., articulação *a posteriori* de preferências, pois não se conhece as preferências *a priori* para considerá-las como dados de entrada. Assim, ao final do processamento, a abordagem deve oferecer um conjunto de soluções viáveis para o problema-alvo, para que dentre as soluções viáveis uma solução fosse selecionada (articulação de preferência *a posteriori*).

Capítulo 4

Experimentos

Esse capítulo descreve os experimentos para a avaliação das abordagens evolucionistas: algoritmo genético multiobjetivo e algoritmo de evolução diferencial multiobjetivo. Tais abordagens estão acopladas a um solucionador de problemas baseados em casos canônico. Deve-se notar que a abordagem baseada em casos encerra uma necessidade importante na estruturação do reuso de experiências; no caso em particular, no reuso de planos de conduções de trens interurbanos de carga. Assim, o conjunto destas abordagens permitiu a confecção de um método multiobjetivo para a geração de planos de condução de trens interurbanos de carga.

Tal método foi avaliado por meio de diferentes experimentos no contexto do domínio de condução de trens interurbanos de carga; como já dito anteriormente, tal avaliação se encerra em ambiente computacional de simulação. Este domínio de aplicação foi utilizado para ilustrar as contribuições da aprendizagem em torno da abordagem de raciocínio baseado em casos na criação de um sistema capaz de: (i) aprender com as experiências; (ii) gerar políticas de condução de trens, e (iii) aplicar tais políticas. Deve-se notar que o sistema resultante permite reusar, de forma transparente, experiências de maquinistas e de condutores de software.

Os experimentos foram modelados sob a forma de um sistema de condução, cf. as seções 2 e 3. E os modelos matemáticos seguem as equações de (LOUMIET, JUNGBAUER e ABRAMS, 2005) (CHANDRA e AGARWAL, 2007). Ao longo desta tese, parte dos resultados foram publicados em (SOUZA, SATO, *et al.*, 2016).

4.1. Ambiente de Execução dos Experimentos

Os experimentos foram realizados em uma máquina contendo 4Gb de memória RAM e com 1 processador Intel Core i5 com 1.80GHz.

4.2. Cenários dos Experimentos

Os experimentos foram definidos para permitir, de um lado, comparar os resultados obtidos das abordagens evolucionistas, a saber: genética e diferencial evolutiva, operando em cenários de otimização multiobjetivo—tempo de viagem e consumo de combustível. E de outro lado, comparar as taxas de acerto das tarefas de recuperação e adaptação de casos considerando as mesmas abordagens evolucionistas supracitadas. Para ambas as abordagens foi utilizada uma população inicial de 50 indivíduos, critério de parada de 10 gerações, probabilidade de mutação de 5% e constante de mutação de 1 para a abordagem diferencial evolutiva. Para embasar estas escolhas foram utilizados os trabalhos relacionados apresentados no Capítulo 2, apenas o critério de parada que optou-se por uma quantidade pequena de gerações, por uma limitação de tempo, considerando que a adaptação de casos é executada para cada sub-trecho da via.

Foram definidos quatro experimentos. Eles foram realizados utilizando os seguintes parâmetros: 8 trens com configurações diferentes (cf. Tabela 3), trafegando em 2 trechos diferentes (*st1* e *st2*). Estas configurações de trens foram escolhidas por já terem sido utilizadas em outros trabalhos (SATO, BORGES, *et al.*, 2012), (BORGES, 2015), (BORGES, DORDAL, *et al.*, 2015), e assim ser possível a comparação com estes trabalhos.

Tabela 3. Configuração de trens dadas pela quantidade de locomotivas, quantidade de vagões e peso.

Trem	Locomotivas	Vagões	Peso (toneladas)
1	3	58	6278
2	4	100	6342
3	4	58	6541
4	2	31	3426
5	3	47	5199
6	2	31	3441
7	4	59	6579
8	2	28	3118

Experimento 1: oito trens sobre o trecho st1 e reuso de experiências

Esse primeiro experimento foi realizado usando as 8 configurações de trens interurbanos de carga definidas anteriormente (cf. Tabela 3). Cada trem realizou uma viagem completa sobre o trecho *st1*. A cada viagem realizada, os casos gerados eram inseridos na base de casos. Desta forma, o trem 2 no início de sua viagem teria a sua disposição como experiência inicial, a base de casos inicial—dada pelo projetista do sistema—mais os casos gerados durante a viagem do trem 1, e assim sucessivamente. O objetivo deste experimento era verificar se haveria economia de consumo de combustível e tempo de viagem, e também validar se as taxas de acerto de recuperação e adaptação dos casos seriam afetadas pelo reuso das experiências.

Experimento 2: oito trens sobre o trecho st2 e reuso de experiências acumuladas

Esse experimento também foi realizado usando as 8 configurações de trens interurbanos de carga definidas anteriormente (cf. Tabela 3). Cada trem realizou uma viagem completa sobre o trecho *st2*, tendo como experiência inicial, a base de casos inicial—dada pelo projetista do sistema—mais os casos gerados durante o Experimento 1. E para cada nova viagem, a experiência do sistema era atualizada com os casos gerados na viagem anterior.

Os experimentos foram projetados a fim de avaliar um cenário realista, em que em um trecho bem-conhecido é explorado usando diversas configurações de trens reproduzem minimamente exemplos diariamente de operação. Assim, o Experimento 1 avalia como as experiências de conduções de trens interurbanos de carga com configurações diferentes podem ser úteis e reaproveitadas por trens com configurações diferentes em viagens futuras. O Experimento 2 avalia o impacto da experiência acumulada em um dado trecho de via férrea usado em outro trecho de via férrea. Qual é o comportamento do sistema conduzindo os mesmos trens em um trecho diferente daquele das suas últimas experiências?

Experimento 3: dois trens semelhantes e reuso de experiências

Esse experimento foi realizado usando as configurações de dois de trens interurbanos de carga definidas anteriormente (cf. Tabela 3), a saber: Trem 1 e Trem 3. Cada trem deste experimento realizou 1 viagem completa sobre o trecho de via férrea *st1*. A execução do experimento iniciou com a viagem do Trem 1. Para tal viagem, o sistema de geração de planos, contou apenas com as experiências relativas a base de casos inicial—aquela dada pelo projetista do sistema. A

execução da segunda viagem feita pelo Trem 3, o sistema de geração de planos contou com as experiências relativas a base de dados inicial—aquela dada pelo projetista do sistema—acrescida das experiências/casos gerados a partir da execução da viagem do Trem 1. Este experimento foi projetado afim de validar a hipótese de que trens com configurações semelhantes, um deles fazendo o reuso das experiências do outro, teriam maior êxito na recuperação e adaptação de casos.

Experimento 4: dois trens diferentes e reuso de experiências

Esse experimento foi realizado usando as configurações de dois de trens interurbanos de carga definidas anteriormente (cf. Tabela 3), a saber: Trem 2 e Trem 8. Cada trem deste experimento realizou 1 viagem completa sobre o trecho de via férrea *sf1*. A execução do experimento iniciou com a viagem do Trem 2. Para tal viagem, o sistema de geração de planos, contou apenas com as experiências relativas a base de dados inicial—aquela dada pelo projetista do sistema. A execução da segunda viagem feita pelo Trem 8, o sistema de geração de planos contou com as experiências relativas a base de dados inicial—aquela dada pelo projetista do sistema—acrescida das experiências/casos gerados a partir da execução da viagem do Trem 2.

Deve-se salientar que todos os experimentos supracitados foram realizados utilizando-se das abordagens evolucionistas: genética e diferencial evolutiva. Ambas abordagens implementaram métodos de otimização multiobjetivo.

4.3. Resultados Obtidos

Nesta seção serão apresentados os resultados obtidos nos experimentos. Eles estão estruturados segundo sua abordagem: genética multiobjetivo e diferencial evolutiva multiobjetivo. E para cada abordagem serão apresentados dados referentes ao desempenho para cada objetivo otimizado, assim como as taxas de acerto das tarefas de recuperação e de adaptação de casos.

4.3.1. Abordagem genética multiobjetivo

Os resultados obtidos foram analisados considerando as seguintes perspectivas de desempenho: economia de combustível, tempo de duração de uma viagem completa, taxa de acerto da tarefa de recuperação e taxa de acerto da tarefa de adaptação.

Os resultados obtidos no Experimento 1 em termos de economia de combustível—medida em LTKB (litro por tonelada bruta transportada)—estão resumidos na Tabela 4 e em termos de tempo de duração de uma viagem completa estão resumidos na Tabela 5. Deve-se notar que os valores obtidos no experimento são confrontados aos valores de desempenho de um maquinista experiente (ser humano). Os valores obtidos mostraram um desempenho bastante satisfatório para abordagem genética. Ela permitiu obter uma economia de combustível de 35.29% para o Trem 2 (menor economia) e de 51.09% para o Trem 1 (maior economia). Ela também permitiu obter uma economia de tempo de duração de viagem completa de 37.44% para o Trem 3 (menor economia) e de 69.50% para o Trem 8 (maior economia).

Tabela 4. Desempenho obtido em termos de LTKB no Experimento 1: usando a abordagem genética multiobjetivo.

Trem	Consumo (LTKB – litro por tonelada bruta transportada)			
	Real (A)	Experimento 1 (B)	Diferença (A-B)	Redução (%)
1	6,19	3,03	3,16	51,09
2	5,68	3,68	2,00	35,29
3	6,23	3,76	2,47	39,62
4	6,49	3,44	3,05	46,95
5	6,29	3,29	3,00	47,68
6	6,17	3,42	2,75	44,53
7	6,26	3,44	2,82	45,05
8	5,68	3,58	2,10	36,98

Tabela 5. Desempenho obtido em termos tempo de duração de viagem completa no Experimento 1: usando a abordagem *genética multiobjetivo*.

Trem	Tempo (minutos)			
	Real (A)	Experimento 1 (B)	Diferença (A-B)	Redução (%)
1	287	116	171	59,53
2	223	91	132	59,18
3	162	101	61	37,44
4	219	109	110	50,02
5	259	120	139	53,76
6	221	93	128	57,71
7	235	103	132	56,03
8	339	103	236	69,50

Importante salientar que para todos os 8 trens houve economia tanto de combustível quanto de tempo.

Nos experimentos 3 e 4 nota-se também um significativo aumento da economia de combustível e redução do tempo de duração de uma viagem completa em relação ao real—executada por um maquinista humano (cf. Tabela 6). No Experimento 3, o maior aumento em termos de econômica de combustível foi de 50.4% para o Trem 1. E a maior redução em termos de tempo de duração de uma viagem completa foi—no Experimento 4—foi de 68.73% para o Trem 8.

Tabela 6. Desempenho obtido em termos de tempo de duração de viagem completa e em termos de LTKB nos Experimentos 3 e 4: usando a abordagem *genética multiobjetivo*

Experimento	Trem	Consumo (LTKB)			Tempo (minutos)		
		Real (A)	Resultado (B)	(A-B) %	Real (A)	Resultado (B)	(A-B) %
3	1	6,19	3,07	50,40	287	118	58,89
3	3	6,23	3,76	39,65	162	99	38,89
4	2	5,68	3,68	35,21	223	90	59,64
4	8	5,68	3,62	36,27	339	106	68,73

Observando a taxa de acerto do processo de recuperação de casos (cf. Gráfico 1), nota-se que houve um aumento da taxa de acerto da recuperação de casos entre as viagens realizadas pelos trens 1 e 3 do Experimento 1. Porém, para os trens 4 e 5 observou-se um declínio. Uma hipótese para isso seria a semelhança das configurações entre o Trem 1 e o Trem 3, e a diferença entre os trens 3, 4 e 5. Comparando os resultados do Experimento 1 com o Experimento 2, observa-se aumento na taxa de acerto da tarefa de recuperação, embora, em alguns trens este

aumento foi muito pequeno. As melhores taxas de acerto obtidas no contexto do Experimento 2 são explicadas pela maior diversidade de experiências a disposição na base de casos.

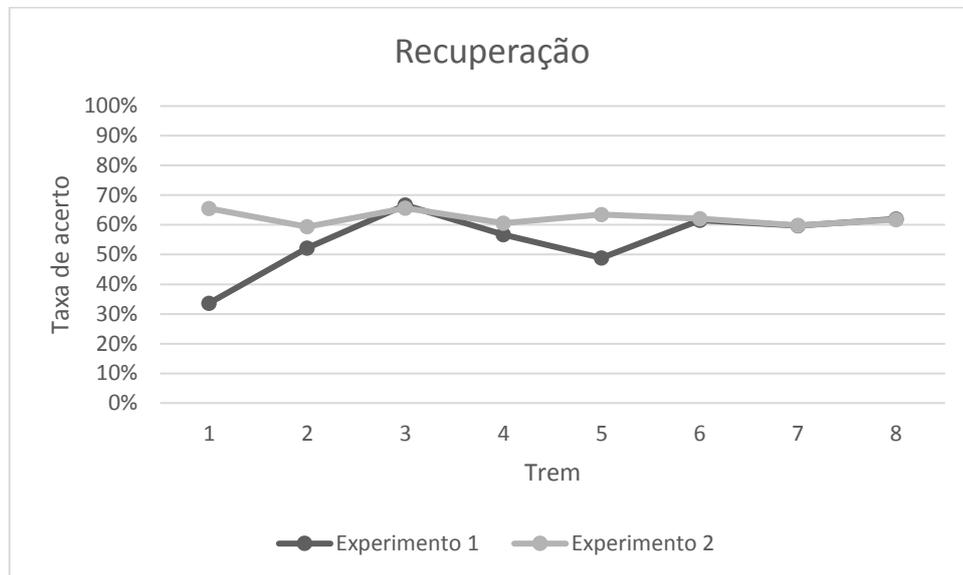


Gráfico 1. Taxa de acerto da tarefa de recuperação de casos: usando a abordagem genética multiobjetivo – Experimentos 1 e 2.

Com relação a taxa de acerto da tarefa de adaptação (cf. Gráfico 2), nota-se que houve um aumento da mesma entre as viagens realizadas pelos trens 1 e 3 do Experimento 1—fato este que também ocorreu com a taxa de recuperação. A explicação para este resultado diz respeito a semelhança entre os trens 1 e 3. Confrontando os resultados obtidos nos experimentos 1 e 2, nota-se que houve um aumento na taxa de acerto da tarefa de adaptação em todos os trens. Este aumento se deve ao reaproveitamento dos casos do Experimento 1, pelo sistema de geração de planos, para gerar as políticas de ações das viagens dos trens do Experimento 2; i.e., mesmo que o trecho de via férrea tenha sido alterado para o Experimento 2, a base de casos enriquecida durante o Experimento 1 traz casos com trens iguais. Idem ao desempenho da tarefa de recuperação, as melhores taxas de acerto aqui obtidas no contexto do Experimento 2 também são explicadas pela maior diversidade de experiências a disposição na base de casos.

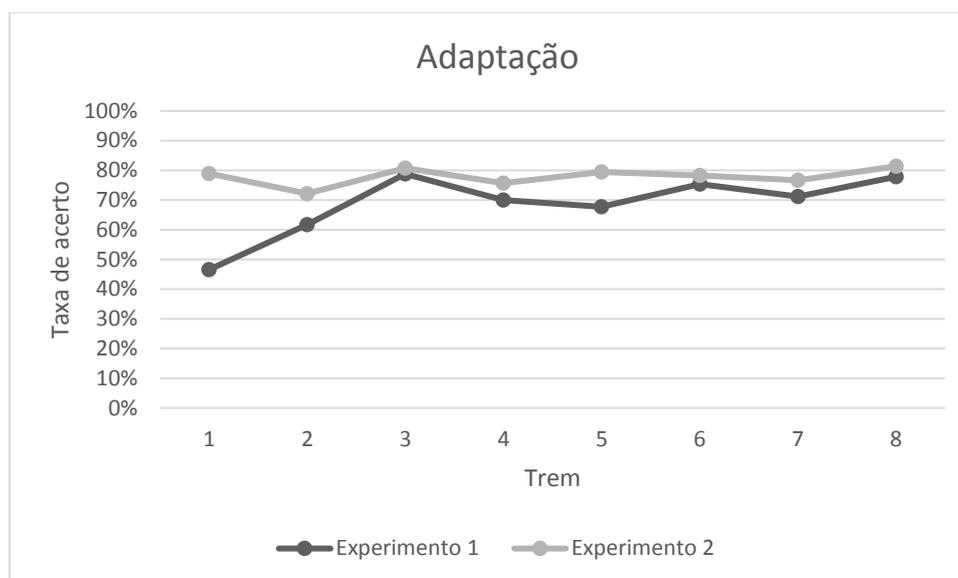


Gráfico 2. Taxa de acerto da tarefa de adaptação de casos: usando a abordagem genética multiobjetivo – Experimentos 1 e 2.

Os resultados obtidos nos Experimentos 3 e 4 são apresentados cf. Tabela 7. Com estes resultados, percebe-se que a hipótese em relação à trens semelhantes é comprovada, ou seja, de que trens com configurações semelhantes, um deles fazendo o reuso das experiências do outro, teriam maior êxito na recuperação e adaptação de casos, do que trens com configurações diferentes (Experimento 4). Aqui, o Trem 1 e o Trem 3 são semelhantes e consequentemente o resultado mostra um aumento nas taxas de acerto de 24.38% de recuperação e de 21.86% na de adaptação. Isto também mostra que a qualidade dos casos gerados durante a viagem do Trem 1 influenciou positivamente na diversidade de planos/experiências reusados no planejamento da viagem do Trem 3. No Experimento 4 nota-se que também houve um aumento nas taxas de acerto, mas esse foi menor: 10.03% na taxa de acerto de recuperação e 8.81% na taxa de acerto de adaptação. Isto se deve ao fato de que no experimento foram usados trens com características muito diferentes. Todavia, deve-se ressaltar que embora os trens tivessem características distintas, ainda assim as taxas de acerto tiveram aumento; mostrando, numa certa medida, a eficiência no modelo proposto para condução automática de trens.

Tabela 7. Desempenho obtido em termos de taxa de acerto das tarefas de recuperação e de adaptação dos Experimentos 3 e 4: usando a abordagem *genética multiobjetivo*.

Taxa de Acerto (%)	Experimento 3			Experimento 4		
	Trem 1 (A)	Trem 3 (B)	Diferença (B-A)	Trem 2 (A)	Trem 8 (B)	Diferença (B-A)
Recuperação de Casos	35,72	60,10	24,38	46,70	56,73	10,03
Adaptação de Casos	48,26	70,12	21,86	54,96	63,77	8,81

4.3.2. *Abordagem diferencial evolutiva multiobjetivo*

Os resultados obtidos foram analisados considerando as seguintes perspectivas de desempenho: economia de combustível, tempo de duração de viagem completa, e taxa de acerto da tarefa de recuperação e taxa de acerto da tarefa de adaptação. Deve-se notar que foram utilizadas aqui as mesmas perspectivas de análise para examinar os resultados obtidos com a *abordagem genética multiobjetivo*.

Os resultados obtidos no Experimento 1 em termos de economia de combustível—medida em LTKB (litro por tonelada bruta transportada)—estão resumidos na Tabela 8 e em termos de tempo de duração de viagem completa estão resumidos na Tabela 9. Deve-se notar que os valores obtidos no experimento são confrontados aos valores de desempenho de um maquinista experiente (ser humano). Os valores obtidos mostraram um desempenho satisfatório para a *abordagem diferencial evolutiva multiobjetivo*—ora em análise. Ela permitiu obter uma economia de combustível de 22.40% para o Trem 2 (menor economia) e de 45.38% para o Trem 1 (maior economia). Ela também permitiu obter uma economia de tempo de duração de viagem completa de 39.37% para o Trem 3 (menor economia) e de 68.11% para o Trem 8 (maior economia). Havendo redução/economia de consumo de combustível e de tempo em todas as viagens realizadas.

Tabela 8. Desempenho obtido em termos de LTKB no Experimento 1: usando a abordagem *diferencial evolutiva multiobjetivo*.

Trem	Consumo (LTKB – litro por tonelada bruta transportada)			
	Real (A)	Experimento 1 (B)	Diferença (A-B)	Redução (%)
1	6,19	3,38	2,81	45,38
2	5,68	4,41	1,27	22,40
3	6,23	4,09	2,14	34,34
4	6,49	3,71	2,78	42,77
5	6,29	3,94	2,35	37,40
6	6,17	3,86	2,31	37,51
7	6,26	3,97	2,29	36,60
8	5,68	4,30	1,38	24,25

Tabela 9. Desempenho obtido em termos de tempo de duração de viagem completa no Experimento 1: usando a abordagem *diferencial evolutiva multiobjetivo*.

Trem	Tempo (minutos)			
	Real (A)	Experimento 1 (B)	Diferença (A-B)	Redução (%)
1	287	114,44	172,56	60,13
2	223	89,87	133,13	59,70
3	162	98,23	63,77	39,37
4	219	105,31	113,69	51,91
5	259	113,47	145,53	56,19
6	221	90,72	130,28	58,95
7	235	102,39	132,61	56,43
8	339	108,10	230,90	68,11

Os experimentos 3 e 4 também aumentaram a economia de combustível e reduziram o tempo de duração de viagem completa em relação ao real (cf. Tabela 10). No Experimento 3, o maior aumento, em termos de economia de combustível, foi observado na viagem realizada pelo Trem 1 com 44.6%. E, no Experimento 4, a maior redução, em termos de tempo de duração de viagem completa, foi observada na viagem realizada pelo Trem 8 com 70.51%.

Tabela 10. Desempenho obtido em termos de tempo de duração de viagem completa e em termos de LTKB nos Experimentos 3 e 4: usando a abordagem *diferencial evolutiva multiobjetivo*.

Experimento	Trem	Consumo (LTKB)			Tempo (Minutos)		
		Real (A)	Resultado (B)	(A-B) %	Real (A)	Resultado (B)	(A-B) %
3	1	6,19	3,43	44,60	287	115,24	59,85
3	3	6,23	4,08	34,44	162	100,99	37,66
4	2	5,68	4,22	25,65	223	91,35	59,04
4	8	5,68	3,88	31,69	339	99,96	70,51

Observando a taxa de acerto da tarefa de recuperação de casos (cf. Gráfico 3), nota-se que não houve diminuição significativa da taxa de acerto da tarefa de recuperação de casos entre os experimentos 1 e 2, exceto para a viagem referente ao Trem 8 do Experimento 2. Uma hipótese para tal exceção é que os casos obtidos a partir do Experimento 1 geram uma boa diversidade de casos e eles foram adequados ao reuso no Experimento 2. Ainda, a mudança de via férrea (*st1* para *st2*), afetou apenas marginalmente a taxa de acerto da tarefa de recuperação. Mas, as taxas de acerto foram consideradas relativamente altas se comparadas a abordagem genética, a menor taxa de acerto na recuperação para o Experimento 1 foi de 78.97% e para o Experimento 2 foi de 88.13%, e as maiores taxas de acerto foram de 83.59% para o Experimento 1 e de 92.31% para o Experimento 2 (cf. Tabela 14).

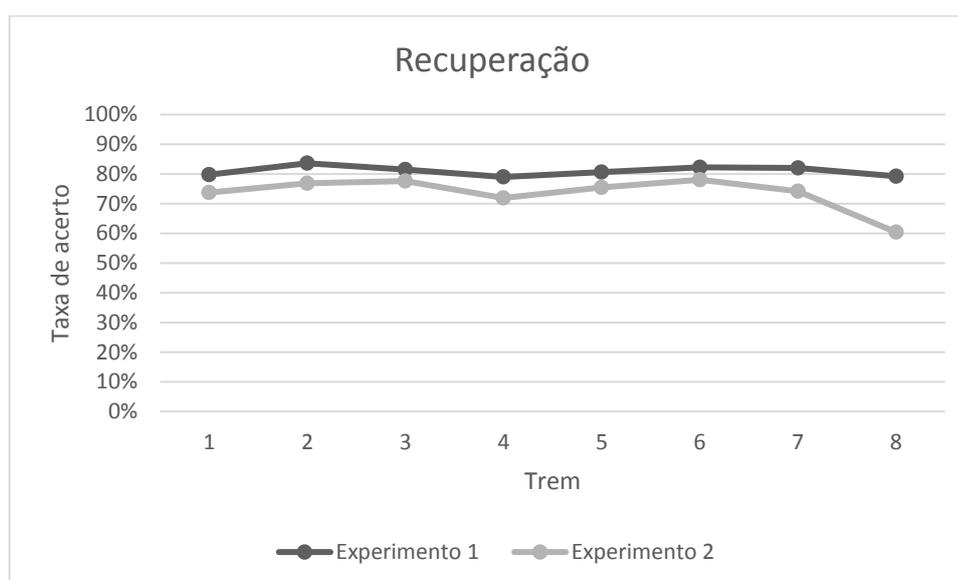


Gráfico 3. Taxa de acerto da tarefa de recuperação de casos: usando a abordagem *diferencial evolutiva multiobjetivo*.

Com relação a taxa de acerto da tarefa de adaptação (cf. Gráfico 4), nota-se que houve uma pequena diferença entre as taxas de acerto da tarefa de adaptação de casos entre a viagem realizada pelos trens 1, 2, 3, 4, 5, 6 e 7. A explicação para esta situação é que as qualidades dos planos adaptados/armazenados com a abordagem *diferencial evolução multiobjetivo* foram superiores aos gerados pela abordagem *genética multiobjetivo*, ou seja, a abordagem *diferencial evolução multiobjetivo* se mostrou bastante eficiente considerando as taxas de acerto para a tarefa de adaptação de casos. Porém, para o Trem 8 houve queda acentuada na taxa de acerto.

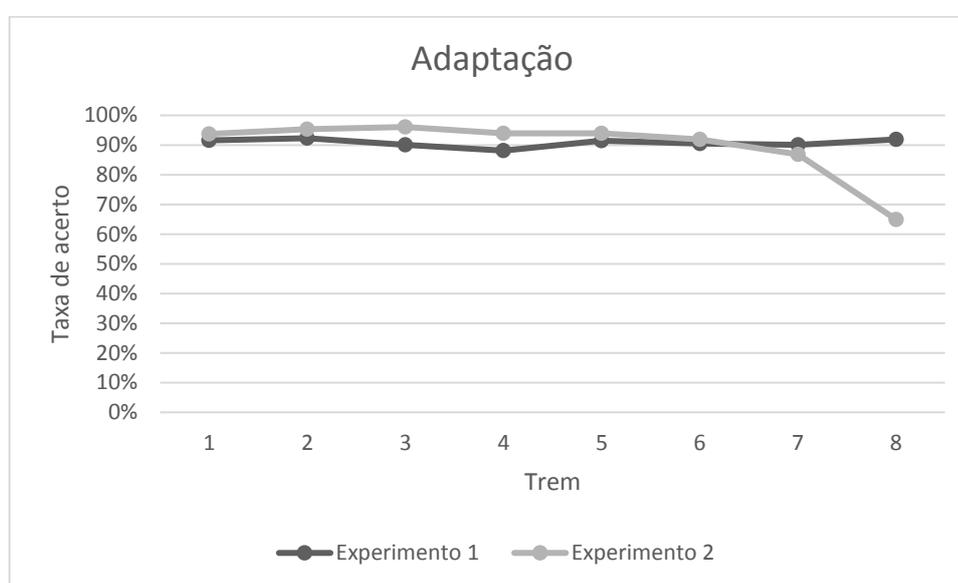


Gráfico 4. Taxa de acerto da tarefa de adaptação de casos: usando a abordagem *diferencial evolutiva multiobjetivo* – Experimentos 1 e 2.

Os resultados obtidos nos experimentos 3 e 4 são apresentados cf. Tabela 11.

Tabela 11. Desempenho obtido em termos taxa de acerto das tarefas de recuperação e adaptação dos Experimentos 3 e 4: usando a abordagem *diferencial evolutiva multiobjetivo*.

Taxa de Acerto (%)	Experimento 3			Experimento 4		
	Trem 1 (A)	Trem 3 (B)	Diferença (B-A)	Trem 2 (A)	Trem 8 (B)	Diferença (B-A)
Recuperação de Casos	79,89	81,81	1,92	83,43	80,95	-2,48
Adaptação de Casos	91,09	91,25	0,16	90,49	89,78	-0,71

Com estes resultados percebe-se que houve pouco aumento no Experimento 3—aquí os trens são semelhantes—e uma pequena diminuição da taxa de acerto no Experimento 4—aquí

os trens são bastante diferentes—, com isto, conclui-se que nestes experimentos as configurações dos trens tiveram pouca influência, embora, no Experimento 3 a semelhança dos trens acarretou em um pequeno aumento da taxa de acerto. E no Experimento 4 as configurações diferentes dos trens, podem ter influenciado na queda da taxa de acerto.

4.4. Discussão dos Resultados

Para verificar o desempenho do modelo CBR utilizando uma abordagem de otimização multiobjetivo, foram comparados os resultados de consumo de combustível com abordagens previamente implementadas descritas em (SOUZA, SATO, *et al.*, 2016) e em (SATO, BORGES, *et al.*, 2012). Este trabalho apresentou e comparou algumas abordagens para a condução automática de trem: aprendizado de máquina, otimização de restrição distribuída, regras especializadas. A Tabela 12 e a Tabela 13 mostram os resultados de economia no consumo de combustível de todas essas abordagens e os modelos de CBR propostos: abordagem *genética multiobjetivo* e abordagem *diferencial evolutiva multiobjetivo*. As colunas ilustram a diferença em porcentagem da economia no consumo de combustível relacionado ao consumo real para cada configuração do trem.

Tabela 12. Desempenho (em LTKB) de diferentes abordagens (trabalhos anteriores) quanto à economia de combustível (em LTKB): CBR+AG (CBR com genética multiobjetivo), CBR+ED (CBR com evolução diferencial multiobjetivo), CBR (CBR com algoritmo genético com soma ponderada), AM: (aprendizagem de máquina), DCOP (otimização de restrição distribuída), SE (sistema baseado em regras), SH (maquinista ser humano).

<i>Trem</i>	<i>CBR+AG</i>	<i>CBR+ED</i>	<i>CBR</i>	<i>AM</i>	<i>DCOP</i>	<i>SE</i>	<i>SH</i>
1	3,03	3,38	3,09	8,91	4,16	5,05	6,19
2	3,68	4,41	3,99	5,77	4,19	5,06	5,68
3	3,76	4,09	3,67	5,71	4,10	4,44	6,23
4	3,44	3,71	3,49	6,20	4,51	6,31	6,49
5	3,29	3,94	3,19	5,93	4,23	5,98	6,29
6	3,42	3,86	3,67	6,10	4,00	5,53	6,17
7	3,44	3,97	3,64	6,05	4,08	5,55	6,30
8	3,58	4,30	4,13	6,38	4,41	6,45	6,26
Média	3,46	3,96	3,61	6,38	4,21	5,55	6,20

Tabela 13. Desempenho de diferentes abordagens (trabalhos anteriores) quando à economia de combustível (em %): CBR+AG (CBR com genética multiobjetivo), CBR+ED (CBR com evolução diferencial multiobjetivo), CBR (CBR com algoritmo genético com soma ponderada), AM: (aprendizagem de máquina), DCOP (otimização de restrição distribuída), SE (sistema baseado em regras), SH (maquinista ser humano).

Trem	CBR+AG (%)	CBR+ED (%)	CBR (%)	AM (%)	DCOP (%)	SE (%)	SH (LTKB)
1	51,05	45,38	50,08	-43,94	32,75	18,42	6,19
2	35,21	22,40	29,75	-1,58	26,26	10,92	5,68
3	39,65	34,34	41,09	8,35	34,26	28,73	6,23
4	47,00	42,77	46,22	4,47	30,49	2,77	6,49
5	47,69	37,40	49,28	5,72	32,83	4,93	6,29
6	44,57	37,51	40,52	1,13	35,20	10,37	6,17
7	45,40	37,01	42,22	3,97	35,31	11,90	6,30
8	42,81	31,26	34,03	-1,92	29,51	3,04	6,26
Média	44,17	36,01	41,65	-2,98	32,07	10,63	6,20

Pode-se observar que a abordagem *genética multiobjetivo* e a abordagem *diferencial evolutiva multiobjetivo* foram mais eficientes que os trabalhos anteriores (aprendizagem de máquina, resolução distribuída de restrições e sistema baseado em regras (SATO, BORGES, *et al.*, 2012)) e também mais eficientes que o maquinista (ser humano), considerando a economia de combustível (em LTKB). Com a abordagem *genética multiobjetivo* obteve-se uma economia de 51.05% (maior economia) e com a abordagem *diferencial evolutiva multiobjetivo* uma economia de 45.38% (maior economia).

A abordagem *genética multiobjetivo* e a abordagem *diferencial evolutiva multiobjetivo* apresentam diferentes taxas de acerto para as tarefas de recuperação e adaptação de casos. Elas podem observadas por meio das curvas nos gráficos (cf. Gráfico 5 e cf. Gráfico 6), podendo-se verificar que a abordagem *diferencial evolutiva multiobjetivo* obteve taxa de acerto maior tanto na tarefa de recuperação quanto na tarefa adaptação de casos.

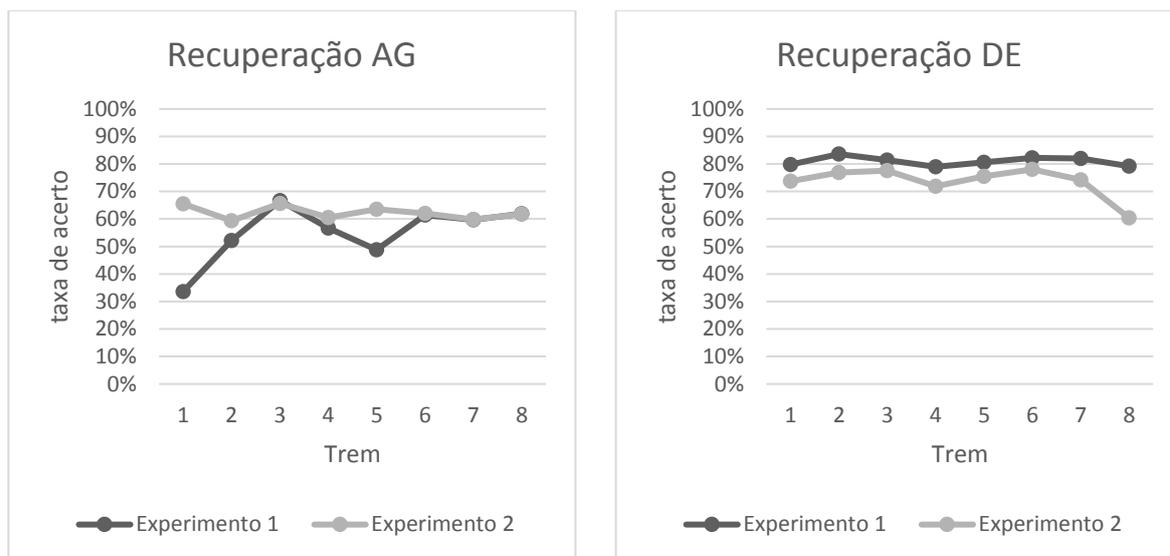


Gráfico 5. Taxa de acerto da tarefa de recuperação de casos: usando as abordagens *genética multiobjetivo* (AG) e *diferencial evolutiva multiobjetivo* (DE) – Experimentos 1 e 2.

Deve-se frisar, no contexto do Experimento 2, que a taxa de acerto da tarefa de recuperação de casos com abordagem *diferencial evolutiva multiobjetivo* foi bastante superior a abordagem *genética multiobjetivo*, atingindo 92.31% para o Trem 2 (Experimento 2) e 88.13% para o Trem 4 (Experimento 2). A taxa de acerto da tarefa de adaptação de casos também foi bastante superior, atingindo 96.10% para o Trem 3 (Experimento 2). Todavia, foi observado um comportamento diferente para o Trem 8, em que a abordagem genética foi superior, isto ocorreu porque o Trem 8 possui o menor peso e a menor quantidade de vagões, e sendo assim, os planos adaptados anteriormente não atenderam esta configuração.

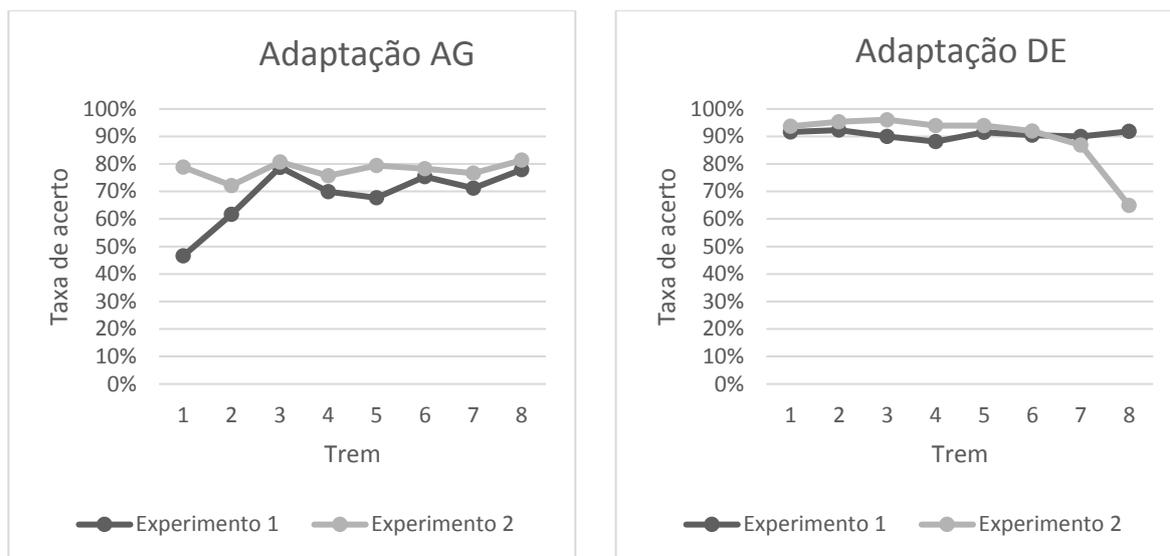


Gráfico 6. Taxa de acerto da tarefa de adaptação de casos: usando as abordagens *genética multiobjetivo* (AG) e *diferencial evolutiva multiobjetivo* (DE) – Experimentos 1 e 2.

A Tabela 14 mostra em detalhe os resultados obtidos, em termos de taxa de acerto das tarefas de recuperação e adaptação de casos, para cada trem e experimento. Analisando os resultados obtidos para ambas as abordagens, percebe-se que os experimentos foram realizados sobre a mesma base de casos inicial, a abordagem diferencial evolutiva apresentou valores para as taxas de acerto tanto de recuperação quanto de adaptação superiores a abordagem genética, então pode-se afirmar que a abordagem diferencial evolutiva tem um aprendizado melhor que a abordagem genética, na medida que ela garantiu a recuperação, adaptação e armazenamento de planos com maior qualidade. Para corroborar tal afirmativa, observar a coluna da Tabela 14, em que no Experimento 1 para ambas as abordagens, usando apenas a base de casos inicial, a taxa de acerto da recuperação de casos da abordagem genética foi de 33.64%, enquanto da abordagem diferencial evolutiva foi de 79.76%.

Tabela 14. Desempenho obtido em termos taxa de acerto das tarefas de recuperação e adaptação dos Experimentos 1 e 2: comparativo das duas abordagens: *abordagem genética (AG)* e *abordagem diferencial evolutiva (DE)*.

Etapa	Abordagem	Trem 1 (%)	Trem 2 (%)	Trem 3 (%)	Trem 4 (%)	Trem 5 (%)	Trem 6 (%)	Trem 7 (%)	Trem 8 (%)
Recuperação de Casos	Exp. 1 - AG	33,64	52,21	66,66	56,67	48,82	61,50	59,75	61,96
	Exp. 1- DE	79,76	83,59	81,43	78,97	80,56	82,27	82,02	79,18
Adaptação de Casos	Exp. 1 – AG	46,58	61,74	78,80	69,95	67,70	75,40	71,23	77,89
	Exp. 1 – DE	73,76	76,89	77,63	71,85	75,47	78,05	74,21	60,39
Recuperação de Casos	Exp. 2 – AG	65,50	59,36	65,64	60,56	63,51	62,05	59,87	61,78
	Exp. 2 – DE	91,63	92,31	90,03	88,13	91,49	90,48	90,04	91,88
Adaptação de Casos	Exp. 2 – AG	78,87	72,18	80,75	75,74	79,44	78,31	76,70	81,36
	Exp. 2 – DE	93,74	95,32	96,10	93,95	93,90	91,92	86,86	64,98

Foi utilizado *Nemenyi e Friedman* (CORDER e FOREMAN, 2009) (FRIEDMAN, 1937) com $\alpha = 0,05$ para validar os resultados de consumo de combustível (em LTKB). Para executar o teste foram utilizados os dados apresentados na Tabela 12. O objetivo foi verificar se existia diferença significativa entre as abordagens que foram desenvolvidas, sendo elas: abordagem genética multiobjetivo e abordagem diferencial evolutiva multiobjetivo e as abordagens de trabalhos anteriores, sendo elas: CBR com algoritmo genético com soma ponderada (BORGES, 2015), aprendizagem de máquina, DCOP (otimização de restrição distribuída), sistema baseado em regras e maquinista (ser humano) (SATO, BORGES, *et al.*, 2012). O teste mostrou que existe uma diferença significativa entre a abordagem genética e aprendizagem de máquina, sistema baseado em regras e maquinista ser humano. Não havendo diferença significativa entre a abordagem genética, abordagem diferencial evolutiva, CBR com algoritmo genético com soma ponderada e DCOP (otimização de restrição distribuída). O mesmo teste foi realizado, mas desta vez usamos apenas os resultados obtidos de consumo de combustível (em LTKB) das abordagens genética multiobjetivo e CBR com algoritmo genético com soma ponderada (BORGES, 2015), abordagem diferencial evolutiva multiobjetivo. O teste mostrou que não há diferença significativa entre as abordagens genéticas multiobjetivo e soma ponderada. Todavia, a abordagem multiobjetivo deve ser privilegiada por representar em termos de definição e modelagem do problema uma visão mais consistente e clara.

4.5. Considerações do Capítulo

A partir dos resultados obtidos pode-se observar que ambas as abordagens: abordagem genética e abordagem diferencial evolutiva utilizadas neste trabalho atenderam e superaram as expectativas, considerando que a economia de consumo de combustível esperada, **inicialmente**, era de 6%, e obtivemos 51.09% na abordagem genética e 45.38% na abordagem diferencial evolutiva. E além de superar a expectativa inicial, as abordagens utilizadas também superaram as demais abordagens de não evolucionistas—exceto DCOP—realizados em nosso Laboratório de Pesquisa de Agentes de Software (LAS).

Com relação a duração das viagens ambas as abordagens permitiram reduzir o tempo de uma viagem completa em no mínimo 37.44% para a abordagem genética multiobjetivo e no mínimo de 39.37% para a abordagem diferencial evolutiva multiobjetivo.

Sobre o ciclo CBR, considerando as etapas de recuperação e adaptação de casos percebe-se que as taxas de acerto foram expressivas. As taxas de acerto na tarefa de recuperação de casos da abordagem diferencial evolutiva multiobjetivo foi bastante superior a abordagem genética multiobjetivo, tanto no Experimento 1 quanto no Experimento 2. O mesmo ocorreu na tarefa de adaptação de casos, exceto para o Trem 8—a configuração deste trem é substancialmente diferente dos demais—, em que a abordagem genética multiobjetivo obteve taxa de acerto maior que a abordagem diferencial evolutiva multiobjetivo.

Conclusão e trabalhos futuros

Este trabalho teve como objetivo geral definir métodos de adaptação de casos com otimização multiobjetivo. Neste contexto foi usado o modal férreo, mais especificamente um cenário realista de condução de trens de carga interurbanos para aplicar os métodos de adaptação de casos com característica multiobjetivo. Para tal, utilizou-se de um simulador computacional de condução automática de trens de carga, em que o mesmo opera a geração de planos de condução de trens de carga interurbanos sobre ciclo clássico de CBR.

Considerando que o cenário de condução de trens apresentava características de problemas em que a otimização multiobjetivo poderia ser aplicada, foi realizado um estudo sobre as diferentes técnicas e métodos de otimização objetivo. Como algoritmo genético com soma ponderada na função de *fitness* (BORGES, 2015) já havia sido utilizada em um trabalho anterior com êxito, optou-se pela abordagem genética multiobjetivo. Como a primeira opção era um algoritmo evolutivo, buscou-se métodos com as mesmas características. Assim, chegou-se a escolha da abordagem evolutiva diferencial, considerando ser um algoritmo relativamente simples e com uma melhor desempenho computacional que os demais métodos de base populacional. Outros fatores que foram considerados dizem respeito às características do simulador usado e as necessidades a serem atendidas, e assim foram avaliados algoritmos de otimização multiobjetivo que permitissem adaptar eficientemente planos de ações. Desta maneira, este trabalho propiciou a geração de planos de ações eficientes para conduzir trens interurbanos de carga com otimização multiobjetivo, sem a intervenção humana.

Deve-se frisar que também foram realizados esforços de pesquisa e desenvolvimento para utilizar *physical programming* juntamente com a abordagem diferencial evolutiva. Porém, não foi possível devido aos pontos de aceleração dos trens serem valores inteiros, e variarem de -1 até 8, pois na *physical programming* é necessário determinar 5 intervalos de valores de acordo com as preferências do usuário, e cenário de condução de trens interurbanos de carga, a preferência do usuário varia de acordo com vários fatores, como perfil da via, configuração do trem, entre outros.

Para avaliar experimentalmente o desempenho de cada algoritmo multiobjetivo selecionado na tarefa de recuperação e adaptação de planos de ações, foram estabelecidos 4 experimentos. Estes experimentos visaram validar a adaptação de casos com diferentes configurações de trens e de vias férreas, e validar o desempenho considerando a economia de combustível (*em LTKB*) e o tempo de duração de uma viagem de trem completa (*em minutos*). Os experimentos também validaram o desempenho de cada algoritmo selecionado quanto a taxa de acerto na tarefa de recuperação e adaptação de planos. Este desempenho refere-se à qualidade de cada caso recuperado e de cada caso adaptado.

Os experimentos demonstraram uma economia média de consumo de combustível de 44.17% na abordagem genética e de 36.01% na abordagem diferencial evolutiva, em relação ao maquinista (humano). Também demonstraram uma economia em termos de tempo de no mínimo 37.44% na abordagem genética e de no mínimo 39.37% na abordagem diferencial evolutiva, em relação ao maquinista (humano). E com relação a qualidade dos casos recuperados e adaptados, verifica-se que na tarefa de recuperação de casos se chegou a um percentual de 92.31 de taxa de acerto, sendo esta a maior taxa de acerto de recuperação de casos. E na tarefa de adaptação de casos obteve-se 96.10% de taxa de acerto, sendo esta a maior taxa de acerto da adaptação de casos.

Os resultados obtidos mostram que a otimização multiobjetivo—usando a abordagem evolucionista—é um caminho promissor, desde que a geração de planos de ações possa ser feita off-line. Deve-se salientar que a combinação da abordagem evolucionista com a abordagem CBR—reusando experiências ou boas soluções já experimentadas—tende fazer com que adaptação de planos usando um método evolucionista seja mais rápida: tanto para a abordagem genética, quanto para a abordagem diferencial evolutiva. Deve-se salientar que quanto mais elevada for qualidade, menor será o esforço da próxima etapa no processo de geração de um plano de políticas de condução de trens interurbanos de carga.

Um ponto importante para salientar é que em termos de execução (tempo de CPU), a abordagem diferencial evolutiva foi considerada muito mais performática que a abordagem genética, corroborando com a literatura (VESTERSTROM e THOMSEN, 2004); embora esta comprovação não tenha sido um objetivo deste trabalho. Porém, vale ressaltar que considerando um cenário de condução automática de trens real, contar com um algoritmo de adaptação com execução mais rápida, eficiente em termos de consumo e tempo, pode ser uma ótima alternativa.

Assim, uma proposta de trabalho futuro é que o simulador de condução atue em uma arquitetura em camadas, baseado na arquitetura *Assumption* (BROOKS, 1985).

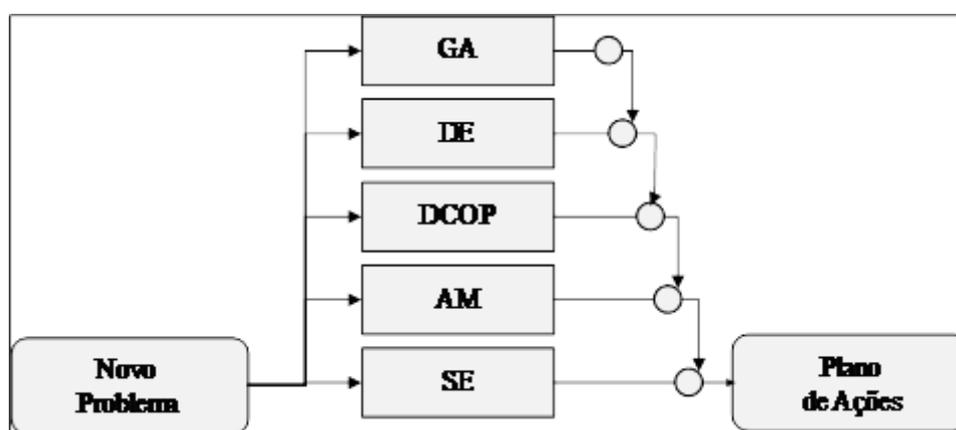


Figura 13. Sistema para a geração de planos de ações estruturado em camadas: SE (sistema baseado em heurísticas), AM (aprendizagem de máquina), DCOP (otimização distribuída de restrições), DE (Evolução Diferencial) e GA (Genética).

Em cada camada uma abordagem de geração de planos de ações (cf. Figura 13). Por exemplo, na base (camada 0) a abordagem mais rápida, e no topo da pilha—última camada—, a abordagem menos rápida e com melhor geração de planos de ações. Assim, se o tempo necessário para se obter um plano é pequeno, o plano será gerado no tempo em questão e com qualidade reduzida. Por outro lado, quanto maior for a disponibilidade de tempo, maiores serão as chances de se obter um plano de ações de melhor qualidade (camada N).

Outra sugestão de trabalho futuro é a adoção de outros métodos de otimização multiobjetivo, tanto evolucionistas, como nuvem de partículas, mas não se limitar a abordagens evolucionistas. Ainda no contexto de otimização multiobjetivo, seria bastante interessante aplicar a otimização multiobjetivo para manter a base de casos representativa e também reduzir seu tamanho, pois durante os experimentos realizados na presente tese notou-se que conforme a base de casos aumentava seu tamanho seu desempenho algumas vezes era afetado. Então, uma otimização multiobjetivo aplicada à manutenção da base de casos mantendo a mesma representativa e performática seria um esforço relevante.

Referências

- AAMODT, A.; PLAZA, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System. **AI Communications. IOS Press**, v. 7, n. 1, p. 39-59, 1994.
- ABRAHAM, ; JAIN, L.; GOLDBERG, R. **Evolutionary Multiobjective Optimization: Theoretical Advances and Applications**. New York: Springer-Verlag, 2005.
- AHMED A. SHABANA, K. E. Z. S. **Railroad Vehicle Dynamics: A Computational Approach**. [S.l.]: CRC Press, 2007.
- AI-JUN, Y. et al. Multi-objective evaluation based optimizing control method for shaft furnace roasting process. **Mechatronics and Automation (ICMA), 2011 International Conference on**, 7-10 Agosto 2011. 967-971.
- ALAM, M. N.; BISWARUP, D.; PANT, V. A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination. **Electric Power Systems Research**, 2015. 39–52.
- AMRANE, Y. et al. Optimal VAR control for real power loss minimization using differential evolution algorithm. **Electrical Power and Energy Systems**, v. 66, p. 262–271, Março 2015. ISSN 0142-0615.
- ATHAN, T. W.; PAPALAMBRO, P. Y. A note on weighted criteria methods for compromise solutions in multi-objective optimization. **Eng. Optim.**, v. 27, p. 155-176, 1996.
- BHADRA, T.; BANDYOPADHYAY, S. Unsupervised feature selection using an improved version of Differential Evolution. **Expert Systems with Applications**, v. 42, n. 8, p. 4042–4053, Maio 2015. ISSN 0957-4174.
- BORGES, A. P. **Descoberta de Regras de Condução de Trens de Carga**. PUCPR. Curitiba, p. 123. 2009.
- BORGES, A. P. **Uma Contribuição Para Geração De Políticas De Ações Para Condução De Trens De Carga Usando Raciocínio Baseado Em Casos.**, Curitiba, 2015.

- BORGES, A. P. et al. **A Multi-Layer Architecture Proposal for Conducting Trains Employing CBR**. Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference. Hsinchu: IEEE. 2014. p. 23-28.
- BORGES, A. P. et al. An Economic Approach for Generation of Train Driving Plans Using Continuous Case-Based Planning. **Proceedings of the 17th International Conference on Enterprise Information Systems**, Barcelona, 1, 26-30 Abril 2015. 440-451. Disponível em: <<http://www.iceis.org>>.
- BRINKER, K.; HÜLLERMEIER, E. Label Ranking in Case-Based Reasoning. In: _____ **Case-Based Reasoning Research and Development**. Northern Ireland, UK: Springer Berlin Heidelberg, 2007. p. 77-91.
- BROOKS, R. A. A robust layered control system for a mobile robot. **Massachusetts Institute of Technology Artificial Intelligence Laboratory**, Massachusetts, v. 864, Setembro 1985.
- CASTRO, R. E. D. **Otimização de Estruturas com Multi-Objetivos Via Algoritmos Genéticos**. Rio de Janeiro. 2001.
- CHANDRA, S.; AGARWAL, M. M. **Railway Engineering**. [S.l.]: Oxford University Press, 2007.
- CHEN, C.-H.; LIU, T.-K.; CHOU, J.-H. Integrated Short-Haul Airline Crew Scheduling Using Multiobjective Optimization Genetic Algorithms. **Systems, Man, and Cybernetics: Systems, IEEE Transactions**, v. 43, n. 5, p. 1077-1090, 2013.
- CHEN, Y.; JIANG, C. Research of the Case Retrieval Model Based on CBR. **Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference**, 16-18 janeiro 2013. 1132-1136.
- CHOU, C.-H.; KUO, B.-H.; CHANG, F. The Generalized Condensed Nearest Neighbor Rule as A Data Reduction Method. **Pattern Recognition, International Conference on, IEEE Computer Society**, v. 2, p. 556-559, 2006.
- COELHO, L. D. S.; ALOTTO, P. Multiobjective Electromagnetic Optimization Based on a Nondominated Sorting Genetic Approach With a Chaotic Crossover Operator. **Magnetics, IEEE Transactions on**, v. 44, n. 6, p. 1078-1081, 2008.

- COELLO, C. A. C. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. **Knowledge and Information Systems**, v. 1, n. 3, p. 269-308, 1999.
- COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. **Evolutionary Algorithms for Solving Multi-Objective Problems**. 2a. ed. New York: Springer, 2007.
- COJAN, J.; LIEBER, J. An Algorithm for Adapting Cases Represented in an Expressive Description Logic. **18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, July 19-22, 2010. Proceedings**, v. 6176, p. 51-65, 2010.
- CORDER, G. W.; FOREMAN, D. I. **Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach**. New Jersey: Wiley, 2009.
- CRAW, S.; MASSIE, S.; WIRATUNG, N. **Informed Case Base Maintenance: A Complexity Profiling Approach**. Fethiye, Turkey: Conference: Advances in Case-Based Reasoning, 8th European Conference, ECCBR 2006. 2006.
- CUMMINS, L.; BRIDGE, D. On Dataset Complexity for Case Base Maintenance. **Case-Based Reasoning Research and Development**, v. 6880, n. Springer, p. 47-61, 2011.
- DAS, I. An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. **Short Paper Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization**, v. 2, p. 411-413, 1999.
- DAS, S.; ABRAHAM, A.; KONAR, A. Differential Evolution Using a Neighborhood-Based Mutation Operator. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 13, n. 3, p. 526-553, Junho 2009. ISSN 1089-778X.
- DAS, S.; SUGANTHAN, N. Differential Evolution: A Survey of the State-of-the-Art. **Evolutionary Computation, IEEE Transactions on**, v. 15, n. 1, p. 4-31, 2011.
- DEB, K. **Multi-objective Optimization using Evolutionary Algorithms**. Nova York: Wiley-Interscience Series in Systems and Optimization, 2001.
- DORDAL, O. B. **Controle, Coordenação e Otimização em Desvios Férreos: Uma Abordagem com Sistema Multiagente Reativo**. PUCPR. Curitiba. 2010.
- DORDAL, O. B. **Uma Contribuição para Condução Férrea Utilizando Agentes Preditivos e Autoconfiguráveis**. PUCPR. Curitiba, p. 169. 2015.

- FANG , W. et al. **Re-scheduling in railway networks**. Computational Intelligence (UKCI), 2013 13th UK Workshop. Guildford: IEEE. 2013. p. 342-352.
- FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. **Journal of the American Statistical Association**, 32, n. 200, 1937. Disponivel em: <<http://dx.doi.org/10.2307/2279372>>.
- GALUSHKA, M.; PATTERSON, D.; NUGENT, C. Assessment of four modifications of a novel indexing technique for case-based reasoning. **International Journal of Intelligent Systems**, v. 22, n. 4, p. 353–383, 2007.
- GARZA, A. D. G. S.; MAHER, M. L. An Evolutionary Approach to Case Adaptation. **Case-Based Reasoning Research and Development**, v. 1650, n. Springer-Verlag Berlin Heidelberg, p. 162-173, 1999.
- GATES , G. W. The Reduced Nearest Neighbor Rule. **IEEE Transactions on Information Theory**, v. 3, n. 431-433, p. 18, 1972.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston: Addison-Wesley Longman Publishing, 1989.
- GRECH, A.; MAIN, J. Case-Base Injection Schemes to Case Adaptation Using Genetic Algorithms. **7th European Conference, ECCBR 2004, Madrid, Spain, August 30 - September 2, 2004. Proceedings**, Agosto 30 2004. 198-210.
- GU , ; CAO, F.; TANG, T. **Energy efficient driving strategy for trains in MRT systems**. Anchorage, AK: IEEE. September 2012. p. 427–432.
- GUAN, D. et al. Nearest neighbor editing aided by unlabeled data. **Information Sciences**, v. 179, n. 13, p. 2273–2282, 2009.
- GUO, Y.; HU, J.; PENG, Y. Research on CBR system based on data mining. **Applied Soft Computing**, v. 11, n. 8, p. 5006-5014, 2011.
- HENGYU , L.; HONGZE, X. **An Integrated Intelligent Control Algorithm for High-Speed Train ATO Systems Based on Running Conditions**. Digital Manufacturing and Automation (ICDMA), 2012 Third International Conference. GuiLin: [s.n.]. 2012. p. 202-205.

- JIAO, L. et al. An Evidential K-Nearest Neighbor Classification Method with Weighted Attributes. **16th International Conference on Information Fusion**, 9-12 Julho 2013. 145-150.
- KANG, Y.-B.; KRISHNASWAMY, S.; ZA, A. A Retrieval Strategy for Case-Based Reasoning Using Similarity and Association Knowledge. **IEEE TRANSACTIONS ON CYBERNETICS**, v. 44, n. 4, p. 473-486, 2014.
- KLARIC, M. N.; SCOTT, G. J.; SHYU, C.-R. Multi-Index Multi-Object Content-Based Retrieval. **IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING**, v. 50, n. 10, p. 4036-4049, 2012.
- KOLODNER, J. L. **Case-based reasoning**. [S.l.]: Morgan Kaufmann Publishers, 1993.
- LEITE, A. R. **Um esquema para redução de consumo de combustível em sistemas de condução férrea baseado em otimização distribuída de restrição**. PUCPR. Curitiba, p. 79. 2009.
- LIAO, Z. et al. Adaptation methodology of CBR for environmental emergency preparedness system based on an improved genetic algorithm. **Expert Systems with Applications**, v. 39, n. 8, p. 7029–7040, 2012.
- LOUMIET, J. R.; JUNGBAUER, W. G.; ABRAMS, B. S. **Train Accident Reconstruction and FELA and Railroad Litigation**. [S.l.]: Lawyers & Judges Publishing Company, 2005.
- LUO, C. et al. **A Computationally Efficient Neural Dynamics Approach to Trajectory Planning of An Intelligent Vehicle**. Neural Networks (IJCNN), 2014 International Joint Conference. Beijing: IEEE. 2014. p. 934-939.
- LUO, Y.; GUIHUA, N. A Niche Hierarchy Genetic Algorithms for Learning Wavelet Neural Networks. **Industrial Electronics and Applications**, Harbin, p. 960-964, Maio 2007. ISSN 978-1-4244-0737-8.
- LUPIANI, E. et al. Case-base maintenance with multi-objective evolutionary algorithms. **Intell Inf Syst**, Nova York, n. 46, 21 Setembro 2016. 259–284.
- MA, S.-X. et al. A case retrieval algorithm based on Ant Colony Clustering. **Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference**, 08-11 Agosto 2009. 39-43.

- MANZOOR, J. et al. Automatic Case Generation for Case-Based Reasoning Systems Using Genetic Algorithms. **Intelligent Systems (GCIS), 2012 Third Global Congress**, 6-8 Novembro 2012. 311-314.
- MARLER, T. R.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. **Structural and Multidisciplinary Optimization**, v. 26, n. 6, p. 369-395, 2004.
- MASSIE, S.; CRAW, S.; WIRATUNGA, N. When Similar Problems Don't Have Similar Solutions. **7th International Conference on Case-Based Reasoning, ICCBR 2007 Belfast**, 13-16 Agosto 2007. 92-106.
- MCDONOUGH, K. et al. **Modeling of Vehicle Driving Conditions Using Transition Probability Models**. Control Applications (CCA), 2011 IEEE International Conference. Denver: IEEE. 2011. p. 544-549.
- MENDES, J. M. A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem. **WSEAS Transactions on Computers**, v. 12, n. 4, p. 164-173, 2013.
- MESSAC, A. Physical programming - Effective optimization for computational design. **AIAA Journal**, v. 34, n. 1, p. 149-158, 1996.
- MESSAC, A.; ISMAIL-YAHAYA, A.; MATTSON, C. A. The normalized normal constraint method for generating the Pareto frontier. **Structural and Multidisciplinary Optimization**, v. 25, n. 2, p. 86-98, 2003.
- MESSAC, A.; MATTSON, C. A. Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming. **Optimization and Engineering**, v. 3, n. 4, p. 431-450, 2002.
- MESSAC, A.; SUKAM, C. P.; MELACHRINOUDI,. Mathematical and Pragmatic Perspectives of Physical Programming. **AIAA JOURNAL**, v. 39, n. 5, p. 885-893, 2001.
- MEZURA-MONTES, E.; COELLO, C. A. C.; TUN-MORALES, E. I. Simple Feasibility Rules and Differential Evolution for Constrained Optimization. **MICAI 2004: Advances in Artificial Intelligence**, Cidade do México, México, n. 2972, p. 707-716, Abril 2004. ISSN 978-3-540-21459-5.

- MINTON, S. Qualitative Results Concerning the Utility of Explanation-Based Learning. **Artificial Intelligence**, v. 42, p. 363-391, 1990.
- MITRA, R.; BASAK, J. Methods of case adaptation: A survey. **International Journal of Intelligent Systems**, v. 20, p. 627-645, 2005.
- MOHSIN, M. F. M. et al. The development of hashing indexing technique in case retrieval. **Information Technology (ITSim), 2010 International Symposium**, v. 2, p. 1045-1050, 2010.
- PERREIRA, A. L. **Estradas - Rodovias e Ferrovias**. 1ª. ed. [S.l.]: Ao Livro Técnico, v. Único, 1958.
- PICEK, S.; GOLUB, M.; JAKOBOVIC, D. Evaluation of Crossover Operator Performance in Genetic Algorithms with Binary Representation. **Bio-Inspired Computing and Applications**, v. 6840, n. 1, p. 223-230, 2012.
- PRENTZAS, J.; HATZILYGEROUDIS, I. Categorizing approaches combining rule-based and case-based reasoning. **Expert Systems**, v. 24, n. 2, p. 97-122, 2007.
- PROFILLIDIS, V. A. **Railway Management and Engineering**. [S.l.]: Ashgate Publishing, Ltd, 2006.
- RAHNAMAYAN, S.; TIZHOOSH, H. R.; SALAMA, M. M. A. Opposition-Based Differential Evolution. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 12, n. 1, p. 64-79, Fevereiro 2008. ISSN 1089-778X.
- RAMÍK, D. M.; SABOURIN, C.; MADANI, K. Autonomous knowledge acquisition based on artificial curiosity: Application to mobile robots in an indoor environment. **Robotics and Autonomous Systems**, v. 61, n. 12, p. 1680-1695, Dezembro 2013.
- RELATÓRIO Anual. **Vale**, 2013. Disponível em: <http://www.vale.com/PT/investors/Quarterly-results-reports/20F/20FDocs/20F_2013_p.pdf>. Acesso em: 15 Abril 2015.
- RELATÓRIO Anual. **Vale**, 2014. Disponível em: <http://www.vale.com/PT/investors/Quarterly-results-reports/20F/20FDocs/Vale%2020-F%202014_p_novo.pdf>. Acesso em: 20 Abril 2015.

- RELATÓRIO Anual. **Vale**, 2016. Disponível em: <http://www.vale.com/PT/investors/information-market/annual-reports/20f/20FDocs/Vale_20-F_FY2016_-_p.pdf>. Acesso em: 06 Fevereiro 2018.
- REZENDE, S. **Sistemas Inteligentes: Fundamentos e Aplicações**. 1a. ed. Barueri, SP: Manole, 2003.
- RITTER, G. L. et al. An algorithm for a selective nearest neighbor decision rule. **IEEE Transactions on Information Theory**, v. 21, p. 665-669, 1975.
- ROSEN, K. H. **Discrete Mathematics and Its Applications**. 1st. ed. [S.l.]: McGraw-Hill, 2012.
- ROSENBERG, R. **Simulation of genetic populations with biochemical properties**. Ann Arbor, MI. 1967.
- ROSENBLOOM, P. S.; GOLDING, A. R. **Improving Accuracy by Combining Rule-based and Case-based Reasoning**. Cambridge, MA. 1995.
- RUNARSSON, T. P.; YAO, X. Stochastic Ranking for Constrained Evolutionary Optimization. **IEEE Transactions on Evolutionary Computation**, 2000. 284-294.
- SABINO, J. A. et al. A multi-objective ant colony optimization method applied to switch engine scheduling in railroad yards. **Revista Brasileira de Pesquisa Operacional**, Rio de Janeiro, v. 30, n. 2, p. 487-514, Maio/Agosto 2010.
- SÁNCHEZA, L.; COUSOB, I.; CASILLA, J. Genetic learning of fuzzy rules based on low quality data. **Fuzzy Sets and Systems**, v. 160, n. 17, p. 2524-2552, 2009.
- SATO, D. M. V. et al. Lessons learned from a simulated environment for trains conduction. **Industrial Technology (ICIT), 2012 IEEE International Conference on**, Athens, 19-21 March 2012. 533-538.
- SCHAFFER , J. D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. **Proceedings of the 1st International Conference on Genetic Algorithms**, p. 93-100, 1985.
- SCHANK, C. R.; RIESBECK, C. K. **Inside Case-based Reasoning**. [S.l.]: Lawrence Erlbaum, 1989.
- SHIU , S. C. K. et al. Maintaining Case-Based Reasoning Systems Using Fuzzy Decision Trees. **Advances in Case-Based Reasoning**, v. 1898, n. 2000, p. 285-296, 2003.

- SILVA, C. J.; NETO, O. M.; MARTINS, F. V. C. Utilização de Algoritmo de Evolução Diferencial Multiobjetivo no Projeto de Controladores. **X Congresso Brasileiro de Inteligencia Computacional (CBIC'2011)**, Fortaleza, Ceará, 8-11 Novembro 2011. 1-7.
- SILVA, M. R. D. **Um Agente Deliberativo Aplicado ao Apoio à Condução de Trens**. PUCPR. Curitiba, p. 101. 2011.
- SMITI, A.; ELOUEDI, Z. Overview of Maintenance for Case based Reasoning Systems. **International Journal of Computer Applications**, v. 32, n. 2, p. 49-56, 2011.
- SMITI, A.; ELOUEDI, Z. Using clustering for maintaining case based. **Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference**, 28-30 Abril 2013. 1-6.
- SMYTH, B.; MCKENNA, E. Modelling the competence of case-bases. **Advances in Case-Based Reasoning**, v. 1488, p. 208-220, 1998.
- SOUZA, V. D. M. et al. Automatic Knowledge Learning Using Case-Based Reasoning. **International Joint Conference on Neural Networks (IJCNN)**, Vancouver, 24-29 Julho 2016. 4579-4585.
- STORN, R.; PRICE, K. Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. **Technical Report**, 1995.
- TSAI, C.-Y.; CHIU, C.-C. Developing a Significant Nearest Neighbor Search Method for Effective Case Retrieval in a CBR System. **International Association of Computer Science and Information Technology - Spring Conference**, 17-10 Abril 2009. 262-269.
- VACEK, S. et al. **Using case-based reasoning for autonomous vehicle guidance**. Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference. San Diego, CA: IEE. 2007. p. 4271-4276.
- VARGAS, D. E. C. et al. Um algoritmo baseado em evolução diferencial para problemas de otimização estrutural multiobjetivo com restrições. **Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería**, v. 32, n. 2, p. 91-99, 2016.

VESTERSTROM, J.; THOMSEN, R. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. **Evolutionary Computation, CEC2004**, v. 2, p. 1980-1987, Junho 2004. ISSN 0-7803-8515-2.

WANGENHEIM, C. G. V.; WANGENHEIM, A. V. **Raciocínio Baseado em Casos**. 1a. ed. [S.l.]: Manole, 2003.

WILSON, D. L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. **IEEE Transactions on Systems Man and Cybernetics**, v. 2, p. 408-421, 1972.

WONG, K. P.; DONG, Z. Y. Differential Evolution, an Alternative Approach to Evolutionary Algorithm. **Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International**, Arlington, VA, p. 73-83, Novembro 2005. ISSN 1-59975-174-7.

YAN, A.; QIAN, L.; ZHANG, C. Memory and forgetting: An improved dynamic maintenance method for case-based reasoning. **Information Sciences**, v. 287, n. 10, p. 50-60, 2014.

ZASLAVSKY, G. M. The simplest case of a strange attractor. **Physics Letters A**, v. 69, n. 3, p. 145-147, Dezembro 1978.

ZHANG, B.; LI, X.; WANG, S. A novel case adaptation method based on an improved integrated genetic algorithm for power grid wind disaster emergencies. **Expert Systems with Applications**, 42, 28 Maio 2015. 7812-7824.

ZHANG, Y.; LOUVIERIS, P.; PETROU, M. Case-Based Reasoning Adaptation for High Dimensional Solution Space. **7th International Conference on Case-Based Reasoning, ICCBR 2007 Belfast, Northern Ireland, UK**, 13-16 Agosto 2007. 149-163.

ZHU, J. **Similarity Metrics and Case Base Maintenance**. [S.l.]. 1998.