



**UNIVERSIDADE ESTADUAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO**

**ALLYSSON ALEX DE PAULA ARAÚJO**

**UMA ARQUITETURA UTILIZANDO ALGORITMO GENÉTICO INTERATIVO E**  
**APRENDIZADO DE MÁQUINA APLICADO AO PROBLEMA DO PRÓXIMO**  
**RELEASE**

**FORTALEZA – CEARÁ**

**2015**

ALLYSSON ALEX DE PAULA ARAÚJO

UMA ARQUITETURA UTILIZANDO ALGORITMO GENÉTICO INTERATIVO E  
APRENDIZADO DE MÁQUINA APLICADO AO PROBLEMA DO PRÓXIMO RELEASE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. PhD. Jerffeson Teixeira de Souza

FORTALEZA – CEARÁ

2015

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

ARAÚJO, Allysson Alex de Paula.

Uma Arquitetura utilizando Algoritmo Genético Interativo e Aprendizado de Máquina aplicado ao Problema do Próximo Release [recurso eletrônico] / Allysson Alex de Paula Araújo. - 2015.

1 CD-ROM: il.; 4 ¼ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 96 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2015.

Área de Concentração: Ciência da Computação

Orientação: Prof. PhD. Jerfesson Teixeira de Souza

1. Problema do Próximo Release. 2. Algoritmo Genético Interativo. 3. Otimização Interativa. 4. Aprendizado de Máquina. 5. Engenharia de Software Baseada em Busca. I. Título.



**UNIVERSIDADE ESTADUAL DO CEARÁ – UECE**  
 PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA - PROPPGq  
 CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT  
 Mestrado Acadêmico em Ciência da Computação – MACC  
 Av. Dr. Silas Munguba, 1700, Campus do Itaperi, - Fone: 3101.9776

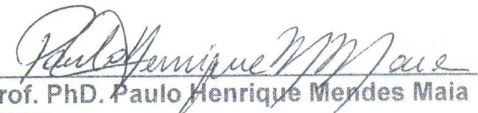


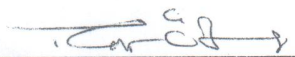
**ATA DA SEPTUAGÉSIMA SEGUNDA DEFESA PÚBLICA  
 DE DISSERTAÇÃO DE MESTRADO**



Aos vinte e três de março de dois mil e quinze, no miniauditório do prédio de Pós-graduação em Computação do Mestrado Acadêmico em Ciência da Computação – MACC, realizou-se a sessão pública de defesa da dissertação de *Allysson Alex de Paula Araújo* aluno regularmente matriculado no Mestrado Acadêmico em Ciência da Computação – MACC, intitulada: *“Uma Arquitetura utilizando Algoritmo Genético Interativo e Aprendizado de Máquina Aplicado ao Problema do Próximo Release”*. A Banca Examinadora reuniu-se no horário de 14:00h às 15:40 horas, sendo constituída pelos Professores da Universidade Estadual do Ceará: **PhD Jefferson Teixeira de Souza** – (Orientador); **PhD Paulo Henrique Mendes Maia e Tibérius de Oliveira e Bonates da Universidade Federal do Ceará-UFC**. Inicialmente o mestrando expôs seu trabalho e a seguir foi submetido à arguição pelos membros da Banca, dispondo cada membro de tempo para tal. Finalmente a Banca reuniu-se em separado e concluiu por considerar o mestrando Aprovado, por sua dissertação e sua defesa pública. Eu, Professor PhD Jefferson Teixeira de Souza Orientador da Dissertação e Presidente da Banca, lavro a presente Ata que será assinada por mim e demais membros. **Fortaleza, 23 de março de 2015.**

  
 Prof. PhD Jefferson Teixeira de Souza  
 Orientador – UECE

  
 Prof. PhD. Paulo Henrique Mendes Maia  
 UECE

  
 Prof. Dr. Tibérius de Oliveira e Bonates  
 UFC

**Cartório Cavalcanti Filho** Rua 7 de setembro, 160 • CEP: 60.720-080 • Parangaba • Fortaleza • Ceará  
 Oficial: JORGE RIBEIRO CAVALCANTI Fone: (35) 3225-0541 • Fax: 3245-1903 • e-mail: ccavalcanti@vetoxmail.com.br

Certifico que esta fotocópia é reprodução fiel do original que me foi apresentado.  
 Fortaleza-CE, 24 de março de 2015-13:59:56. Usuário.: MARCIO

Aquida Maria Pereira de Oliveira -Escrvente Autorizada  
 Emolumentos: R\$ 1,03 Taxas: R\$ 0,82 Total: R\$ 1,85

Válido somente com selo de autenticidade



ALLYSSON ALLEX DE PAULA ARAÚJO

UMA ARQUITETURA UTILIZANDO ALGORITMO GENÉTICO INTERATIVO E  
APRENDIZADO DE MÁQUINA APLICADO AO PROBLEMA DO PRÓXIMO RELEASE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 23 de Março de 2015

BANCA EXAMINADORA

---

Prof. PhD. Jerffeson Teixeira de Souza (Orientador)  
Universidade Estadual do Ceará – UECE

---

Prof. PhD. Paulo Henrique Mendes Maia  
Universidade Estadual do Ceará - UECE

---

Prof. PhD. Tibérius de Oliveira e Bonates  
Universidade Federal do Ceará - UFC

Este trabalho é dedicado à minha família.

## AGRADECIMENTOS

Agradeço primeiramente à Deus por todas as graças alcançadas.

À minha família pelo apoio incondicional, em especial aos meus pais Adriana Sousa de Paula e Hélio Pitombeira de Araújo, minha vó Francisca Teles de Sousa e meu irmão Nicholas Andrey de Paula Araújo.

Ao meu orientador prof. PhD. Jerffeson Teixeira de Souza pela confiança concedida nas mais diversas tarefas e, claro, por todo conhecimento compartilhado. Serei sempre grato pelo suporte imprescindível nos momentos que mais precisei e pela grande amizade construída.

Aos professores Me. Sidney Lima, Dr. Marcelo Moraes e Me. Aristofânio Garcia por todo incentivo no início dessa importante fase da minha vida.

À todos os amigos do Grupo de Otimização em Engenharia de Software (GOES.UECE) pelos momentos de aprendizado e alegria, em especial ao Matheus Paixão pelo fundamental apoio e grande amizade conquistada; ao Thiago Nascimento, Altino Dantas e Ítalo Yeltsin pelas sempre presentes contribuições.

Aos velhos amigos de Juazeiro do Norte, Lászlón Costa, Bruno Brito, Marcelo Peixoto e Weskley Vinícius.

À todos os amigos e professores do Mestrado Acadêmico em Ciências da Computação (MACC), em especial ao professores PhD. Paulo Henrique Mendes Maia e Dr. Gustavo Augusto de Lima Campos pelos ensinamentos e conversas amistosas.

Ao prof. PhD. Tibérius de Oliveira e Bonates por aceitar o convite de participação e contribuição na minha banca avaliadora.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por todo apoio financeiro concedido para desenvolvimento deste mestrado.

“Strength does not come from winning. Your struggles develop your strengths. When you go through hardships and decide not to surrender, that is strength.”

(Arnold Schwarzenegger)



## RESUMO

O Problema do Próximo *Release* consiste na seleção de quais requisitos serão implementados no próximo *release* do sistema. Para muitas abordagens que envolvem *Search Based Software Engineering* e o Problema do Próximo *Release*, ainda falta a habilidade de incluir eficientemente a opinião humana e suas peculiaridades no processo de busca. Assim, neste trabalho apresenta-se uma arquitetura para resolver o Problema do Próximo *Release* que permita incorporar as preferências humanas e que, através de um modelo de aprendizado, esteja apto a minimizar o problema da fadiga humana. Os resultados experimentais são aptos a demonstrar que a abordagem proposta é capaz de incorporar de forma bem-sucedida as preferências do usuário na solução final com uma baixa perda em termos objetivos.

**Palavras-chave:** Problema do Próximo Release. Algoritmo Genético Interativo. Otimização Interativa. Aprendizado de Máquina. Engenharia de Software Baseada em Busca.

## **ABSTRACT**

The Next Release Problem consists in selecting which requirements will be implemented in the next software release. For many Search Based Software Engineering approaches to the Next Release Problem, there is still a lack of ability to efficiently include the human opinion and its peculiarities in the search process. Thus, in this work it is proposed an architecture to solve the Next Release Problem where the human preferences can be incorporated and, through a learning model, is able to minimize the problem of human fatigue. Experimental results are able to show that an Interactive Genetic Algorithm can successfully incorporate the user preferences in the final solution with a small loss in objectives terms.

**Keywords:** Next Release Problem. Interactive Genetic Algorithm. Interactive Optimization. Machine Learning. Search Based Software Engineering.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Número de publicações em SBSE por ano entre 1976 e 2014 . . . . .	22
Figura 2 – Taxa de publicações em SBSE por área entre 1976 e 2014 . . . . .	23
Figura 3 – Sistema CEI Genérico . . . . .	25
Figura 4 – Número de publicações que utilizam conceitos de Otimização Interativa em SBSE entre 2002 e 2014 . . . . .	26
Figura 5 – Taxa de publicações que utilizam conceitos de Otimização Interativa em SBSE por área entre 2002 e 2014 . . . . .	27
Figura 6 – Exemplos de Cruzamento e Mutação . . . . .	31
Figura 7 – Classificação dos Sistemas Aprendizado de Máquina . . . . .	34
Figura 8 – Modelo de um Neurônio . . . . .	36
Figura 9 – Gráfico da função Limiar . . . . .	38
Figura 10 – Gráfico da função Limiar por partes . . . . .	38
Figura 11 – Gráfico da função Sigmóide . . . . .	38
Figura 12 – Interpretação Geométrica do <i>Least Median Square</i> . . . . .	41
Figura 13 – Modelo <i>Multilayer Perceptron</i> . . . . .	43
Figura 14 – Método Iterativo e Incremental . . . . .	45
Figura 15 – Visão Geral da Arquitetura . . . . .	55
Figura 16 – Diagrama de Atividades da Arquitetura . . . . .	56
Figura 17 – Relação entre <i>Grau de Similaridade</i> e Quantidade de Perguntas . . . . .	69
Figura 18 – <i>Fator de Similaridade</i> . . . . .	72
Figura 19 – <i>Preço da Preferência</i> . . . . .	74
Figura 20 – Relação entre Grau de Similaridade e Preço da Preferência . . . . .	76

## LISTA DE TABELAS

Tabela 1 – Valores referentes ao <i>Grau de Similaridade</i> para diferentes quantidades de perguntas utilizando LMS e MLP . . . . .	68
Tabela 2 – Valores referentes ao <i>Grau de Similaridade</i> e <i>Fator de Similaridade</i> ampliando o peso $\beta$ utilizando LMS e MLP . . . . .	71
Tabela 3 – Valores referentes ao <i>score</i> e <i>Preço da Preferência</i> ampliando o peso $\beta$ utilizando LMS e MLP . . . . .	73

## LISTA DE QUADROS

Quadro 1 – Visão Geral do Cenário . . . . .	61
Quadro 2 – Relação entre Quantidade de Perguntas e Instâncias Utilizadas . . . . .	69

## LISTA DE ALGORITMOS

1	Algoritmo Genético Canônico . . . . .	30
2	Cálculo da Aptidão do Indivíduo no IGA . . . . .	59

## LISTA DE ABREVIATURAS E SIGLAS

CE	Computação Evolucionária.
CEI	Computação Evolucionária Interativa.
ERP	Enterprise Resource Planning.
FS	Fator de Similaridade.
GS	Grau de Similaridade.
IGA	Interactive Genetic Algorithm.
iNRP	Interactive Next Release Problem.
LMS	Least Median Square.
MLP	Multilayer Perceptron.
NRP	Next Release Problem.
PP	Preço da Preferência.
RNA	Redes Neurais Artificiais.
SBSE	Search Based Software Engineering.
she	Avaliação Subjetiva Humana.
WEKA	Waikato Environment for Knowledge Analysis.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	17
1.1	MOTIVAÇÃO	17
1.2	OBJETIVOS	18
1.3	ESTRUTURA DO TRABALHO	19
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	21
2.1	ENGENHARIA DE SOFTWARE BASEADA EM BUSCA	21
2.2	OTIMIZAÇÃO INTERATIVA	24
2.3	OTIMIZAÇÃO INTERATIVA EM ENGENHARIA DE SOFTWARE BASE- ADA EM BUSCA	26
2.4	ALGORITMO GENÉTICO INTERATIVO	29
2.5	APRENDIZADO DE MÁQUINA	32
<b>2.5.1</b>	<b>Redes Neurais</b>	35
<b>2.5.2</b>	<b>Técnicas Aplicadas</b>	39
2.5.2.1	Least Median Square	39
2.5.2.2	Multilayer Perceptron	41
<b>2.5.3</b>	<b>Ferramenta WEKA</b>	44
2.6	PROBLEMA DO PRÓXIMO RELEASE	44
2.7	CONCLUSÕES DO CAPÍTULO	48
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	49
3.1	PROBLEMA DO PRÓXIMO RELEASE	49
3.2	OTIMIZAÇÃO INTERATIVA EM ENGENHARIA DE SOFTWARE BASE- ADA EM BUSCA	50
3.3	TRATAMENTO DA FADIGA HUMANA	52
3.4	CONCLUSÕES DO CAPÍTULO	53
<b>4</b>	<b>ARQUITETURA E FORMULAÇÃO INTERATIVA PARA O NRP</b>	54
4.1	VISÃO GERAL DA ARQUITETURA	54
4.2	UMA FORMULAÇÃO INTERATIVA PARA O NRP	57
<b>4.2.1</b>	<b>Modelagem Matemática</b>	57
<b>4.2.2</b>	<b>Modelagem Interativa</b>	60
4.3	EXEMPLO DE APLICAÇÃO	60



4.4	CONCLUSÕES DO CAPÍTULO . . . . .	62
<b>5</b>	<b>ESTUDO EMPÍRICO . . . . .</b>	<b>64</b>
5.1	INSTÂNCIAS . . . . .	64
5.2	CONFIGURAÇÕES . . . . .	64
5.3	MÉTRICAS . . . . .	66
<b>5.3.1</b>	<b>Grau de Similaridade . . . . .</b>	<b>66</b>
<b>5.3.2</b>	<b>Fator de Similaridade . . . . .</b>	<b>66</b>
<b>5.3.3</b>	<b>Preço da Preferência . . . . .</b>	<b>67</b>
5.4	RESULTADOS E ANÁLISES . . . . .	67
<b>5.4.1</b>	<b>Relação entre quantidade de perguntas e preferências atendidas . . . . .</b>	<b>67</b>
<b>5.4.2</b>	<b>Ganho Subjetivo . . . . .</b>	<b>70</b>
<b>5.4.3</b>	<b>Perda Objetiva . . . . .</b>	<b>71</b>
<b>5.4.4</b>	<b>Trade-off entre Ganho Subjetivo e a Perda Objetiva . . . . .</b>	<b>75</b>
5.5	AMEAÇAS À VALIDADE . . . . .	75
5.6	CONCLUSÕES DO CAPÍTULO . . . . .	78
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>80</b>
6.1	CONTRIBUIÇÕES . . . . .	81
6.2	LIMITAÇÕES DA PESQUISA . . . . .	82
6.3	TRABALHOS FUTUROS . . . . .	82
	<b>REFERÊNCIAS . . . . .</b>	<b>84</b>
	<b>APÊNDICES . . . . .</b>	<b>89</b>
	APÊNDICE A – Resultados utilizando LMS . . . . .	90
	APÊNDICE B – Resultados utilizando MLP . . . . .	93

# 1 INTRODUÇÃO

Neste capítulo apresentam-se os fatores que motivaram e, conseqüentemente, justificaram o desenvolvimento desta pesquisa. Durante o mesmo, serão devidamente definidos os objetivos e a estrutura do trabalho.

## 1.1 MOTIVAÇÃO

Durante o processo iterativo e incremental de desenvolvimento de software, existem alguns problemas que são naturalmente complexos, como por exemplo, a seleção de quais requisitos serão implementados no próximo *release* do sistema. O termo “*release*” é utilizado para descrever uma versão estável e executável do sistema que será entregue ao cliente, de acordo com suas respectivas preferências. Tal problema é presente e recorrente durante todo o ciclo de desenvolvimento de sistemas. Diante desse contexto, pode-se mencionar o amplamente conhecido Problema do Próximo *Release*, em inglês *Next Release Problem* (NRP) (BAGNALL et al., 2001), cujo cerne baseia-se na maximização da satisfação do cliente respeitando um *budget*, que se refere a um orçamento previamente definido. Essa satisfação é atendida quando os requisitos estipulados como mais importantes pelo cliente são implementados no próximo *release*.

A principal meta na Engenharia de Software Baseada em Busca, em inglês *Search Based Software Engineering* (SBSE), é reformular algumas das dificuldades que residem na Engenharia de *Software* como problemas de busca. Nesse caso, é válido ressaltar que o termo “busca” remete à utilização de técnicas baseadas em busca, especialmente metaheurísticas, que auxiliam na procura das melhores soluções. As mesmas são guiadas através de uma função de avaliação cuja finalidade consiste em distinguir especificamente quais serão as soluções boas ou ruins (HARMAN et al., 2012). Logo, devido o NRP ser um problema *NP-hard*, a aplicação de metaheurísticas se destaca como uma interessante alternativa de resolução (BAGNALL et al., 2001).

As atuais abordagens de resolução do NRP utilizando os conceitos de SBSE podem ser consideradas como ferramentas de tomada de decisão, onde a seleção de requisitos para o próximo *release* é feita de forma automática, sem a participação efetiva de um ser humano. Conseqüentemente, sabe-se que pode existir uma certa insegurança por parte dos usuários em aceitar resultados dessa natureza. Além disso, existem diversos aspectos intrínsecos ao desenvolvimento

de software cuja compreensão é inerentemente subjetiva. Diante dessas questões, presume-se que incluir o usuário durante o processo de busca é uma estratégia válida para tratar o NRP, haja vista que propiciará à técnica de busca a possibilidade de ser guiada pelo conhecimento humano.

Considerando as dificuldades em incluir o tomador de decisão nos problemas de otimização em geral, pode-se destacar a Otimização Interativa. Essa estratégia visa empregar o ser humano de forma protagonista no processo de avaliação das soluções, de maneira que seu conhecimento e demais aspectos psicológicos sejam devidamente incorporados no processo de busca (HARMAN et al., 2009). Logo, o que faz o ser humano ter um papel proeminente nesse contexto é justamente quando a função de avaliação não pode ser explicitamente capturada através de modelos matemáticos ou quando há exigência de julgamentos pessoais (HARMAN et al., 2012).

Apesar de diferentes versões de Algoritmos Genéticos serem amplamente difundidas em SBSE, ainda falta à maioria de suas aplicações a capacidade de utilizar as preferências humanas no processo de busca. Dessa maneira, o Algoritmo Genético Interativo, em inglês *Interactive Genetic Algorithm* (IGA), destaca-se como uma interessante alternativa para lidar com esse problema. O IGA segue os mesmos conceitos da versão canônica do Algoritmo Genético, exceto no quesito referente à avaliação de soluções, pois nessa abordagem a responsabilidade de julgamento das soluções será transferida ao usuário ao invés de apenas uma função matemática em especial (TAKAGI, 2001).

O intenso envolvimento humano no processo de busca se mostra muito interessante e atrativo, porém, o mesmo também pode resultar em um dos problemas mais críticos nas abordagens que utilizam a Otimização Interativa, que é a fadiga humana (TAKAGI, 2001). Essa exaustão pode ser tanto de natureza física quanto psicológica e ocorre devido as repetidas requisições de avaliações ao usuário durante o processo, o que acaba tornando-se uma grande adversidade para evolução do IGA.

## 1.2 OBJETIVOS

O presente trabalho tem como objetivo principal apresentar uma arquitetura para resolver o Problema do Próximo *Release* que permita incorporar as preferências humanas durante o processo de busca através de um Algoritmo Genético Interativo. Visando minimizar o problema da fadiga humana, propõe-se a utilização de modelo de aprendizado de máquina apto a aprender o comportamento do usuário e posteriormente substituí-lo.

Em termos de objetivos específicos, pretende-se:

- a) Propor uma formulação interativa do NRP que consiga se adequar aos requisitos definidos pela arquitetura proposta;
- b) Elaborar métricas que contribuam na análise de resultados que exploram os conceitos de Otimização Interativa em SBSE;
- c) Testar e analisar o comportamento da arquitetura proposta, incluindo a aplicação de duas técnicas distintas de aprendizado de máquina. Visando facilitar a análise deste objetivo específico, os experimentos realizados objetivam responder quatro questões de pesquisa devidamente descritas durante a Seção 5.4

### 1.3 ESTRUTURA DO TRABALHO

O trabalho está organizado em sete capítulos, incluindo a presente introdução. De maneira geral, o restante dos capítulos são resumidos abaixo:

- a) **Capítulo 2 - Fundamentação Teórica:** são discutidos os principais conceitos que permeiam a pesquisa desenvolvida. Primeiramente, aborda-se os princípios que envolvem a linha de pesquisa reconhecida como Engenharia de Software Baseada Busca. Posteriormente explana-se os fundamentos relacionados à Otimização Interativa, sua aplicação em SBSE e a metaheurística Algoritmo Genético Interativo. Em seguida discute-se os conceitos que envolvem a área de Aprendizado de Máquina, incluindo as técnicas utilizadas no presente trabalho. Por fim, apresentam-se as particularidades referentes ao Problema do Próximo *Release*.
- b) **Capítulo 3 - Trabalhos Relacionados:** são analisados os principais trabalhos relacionados ao Problema do Próximo *Release*, a aplicação dos conceitos de Otimização Interativa em SBSE e ao tratamento da fadiga humana.
- c) **Capítulo 4 - Arquitetura e Formulação Interativa para o NRP:** a princípio define-se todos os componentes e fases que compõe a arquitetura proposta. Subsequentemente, apresenta-se uma formulação interativa para o Problema do Próximo *Release* sob a perspectiva da arquitetura previamente definida. Ao final, descreve-se um possível cenário de aplicação da abordagem.
- d) **Capítulo 5 - Estudo Empírico:** em prol de uma validação e avaliação empírica, a proposta elaborada é submetida a um conjunto de testes. Logo, discute-se as composição das instâncias, configurações, métricas, bem como são analisadas as

conclusões referentes ao comportamento da abordagem. Ao fim, discute-se as ameaças à validade dos experimentos

- e) **Capítulo 6 - Considerações Finais:** são discutidas as últimas considerações, contribuições alcançadas e limitações da pesquisa. Ao final, elenca-se um conjunto de oportunidades para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos os principais conceitos e reflexões necessários para compreensão da abordagem e estudo empírico realizados nesta dissertação. Consequentemente, promove-se um maior embasamento teórico dos temas a serem explorados. À princípio, expõe-se os conceitos primordiais para utilização da Engenharia de Software Baseado em Busca. Posteriormente, são elucidados os fundamentos que envolvem a Otimização Interativa e, em seguida, sua aplicação na Engenharia de Software Baseada em Busca. Após essa explanação, são apresentados os princípios referentes a técnica de busca denominada Algoritmo Genético Interativo. Logo após, discute-se as concepções relacionados a área de Aprendizado de Máquina, detalhando as duas técnicas testadas no presente trabalho. Por fim, apresentam-se as características e motivações que levaram a se optar pelo Problema do Próximo *Release* como estudo de caso para aplicação da arquitetura proposta.

### 2.1 ENGENHARIA DE SOFTWARE BASEADA EM BUSCA

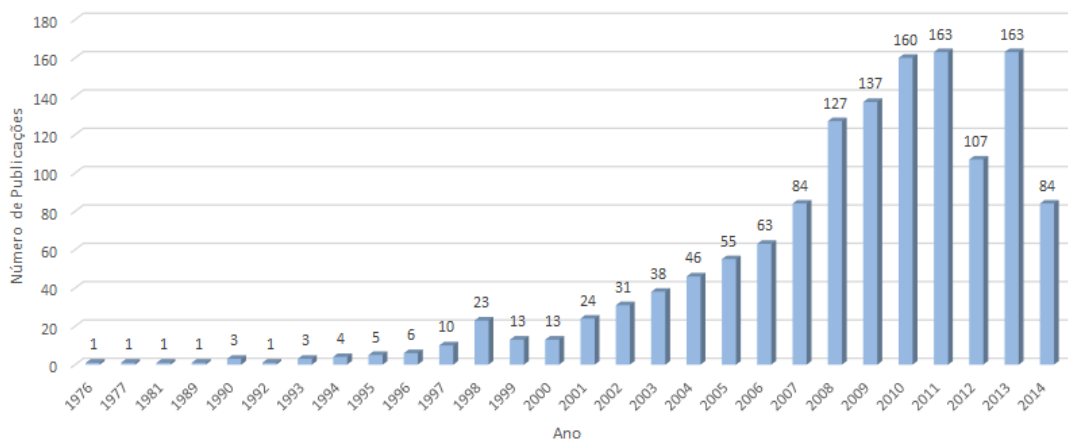
A Engenharia de *Software* possui inerentemente características matemáticas e problemas cuja soluções são difíceis de se alcançar, seja devido a presença de objetivos conflitantes, desconhecimento de soluções ou mesmo a existência de um espaço de busca muito grande. Dessa forma, constata-se o quão difícil é encontrar uma solução ótima para problemas que possuem alta complexidade estrutural (HARMAN; JONES, 2001). Nas diferentes fases da Engenharia de *Software*, muitas situações podem ser modeladas como problemas de otimização e, inclusive, ser resolvidas através de algoritmos de busca. Portanto, em prol de respostas mais adequadas para essas situações, surge a linha de pesquisa denominada como Engenharia de Software Baseada em Busca, em inglês *Search Based Software Engineering* (SBSE).

O termo *Search Based Software Engineering* foi denominado por Harman e Jones (2001) com o propósito de definir a aplicação de técnicas de busca em problemas complexos de Engenharia de *Software* a fim de otimizar (maximizar ou minimizar) uma função ou grupo de fatores que afetam o problema. Ao se modelar matematicamente tais dificuldades, torna-se plausível a resolução dos mesmos através de algoritmos de otimização, dentre os quais destacam-se as metaheurísticas. Apesar de trabalhos nesse contexto, como (MILLER; SPOONER, 1976), foi a partir do artigo publicado por Harman e Jones em 2001 que a pesquisa em SBSE realmente começou a se intensificar e consolidar suas aplicações nas diversas ramificações do desenvolvimento

de software, como por exemplo, priorização de casos de teste de software (ROTHERMEL et al., 2001), planejamento de *releases* (GREER; RUHE, 2004), manutenção de software (O'KEEFFE; CINNÉIDE, 2008), correção automática de *bugs* (WEIMER et al., 2010), entre outros.

Na Figura 1 pode-se constatar tal ascendência ao se visualizar a crescente quantidade de trabalhos produzidos na área, principalmente após o ano de 2001, conforme mencionado anteriormente. Também é válido destacar a importância do SBASE perante a comunidade de SBSE. O mesmo é reconhecidamente o maior repositório de artigos da área e é mantido por Zhang et al. (2011).

Figura 1 – Número de publicações em SBSE por ano entre 1976 e 2014



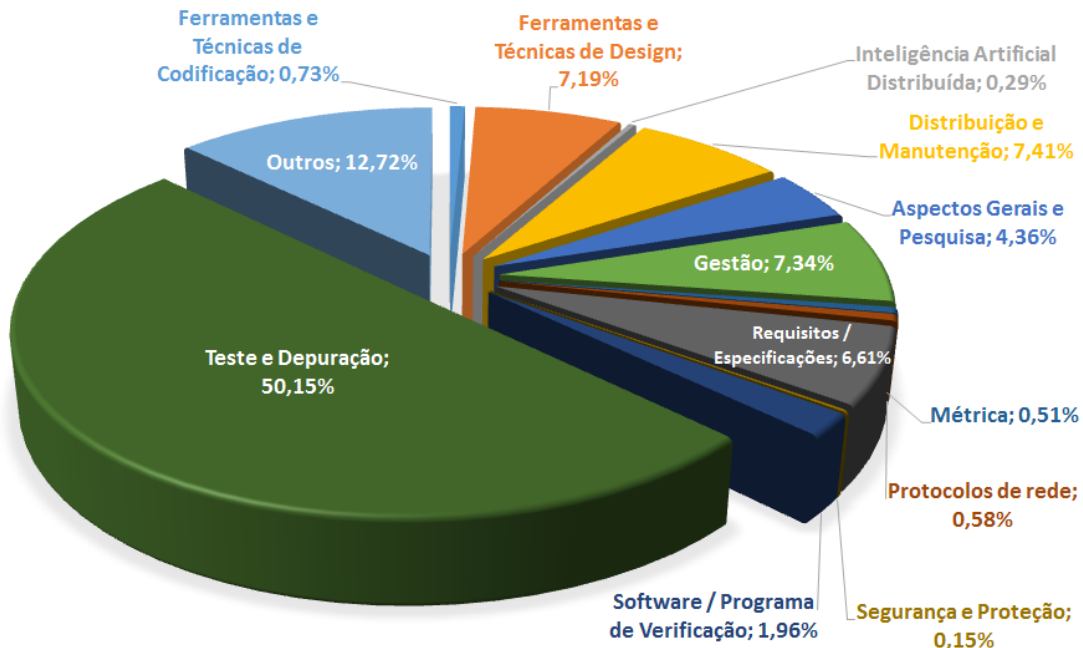
Fonte: Adaptado de Zhang et al. (2011).

Outra análise relevante que pode ser verificada na Figura 2 é a quantidade de publicações por área de pesquisa. É interessante notar que o domínio mais abordado é o de Testes e Depuração, o qual responde por mais de 50% das publicações. Tal valor se justifica devido a natural necessidade de automatização inerentes a área de testes. Requisitos, tema explorado nesta dissertação, conta com uma taxa de 6,61%.

Devido seu amplo grau de generalidade, os algoritmos de busca também se diferenciam como uma interessante alternativa na adaptação de problemas específicos. Todavia, Harman (2007) ressalta que, antes de aplicar as técnicas de busca para determinado problema, é necessário modelar o mesmo através da definição de dois componentes que influenciam diretamente no sucesso do projeto, que são:

- a) **Representação da solução:** reflete como a solução do problema será representada simbolicamente, ou seja, se será constituída por números binários, vetores, matrizes, etc.

Figura 2 – Taxa de publicações em SBSE por área entre 1976 e 2014



Fonte: Adaptado de Zhang et al. (2011).

- b) **Função objetivo:** também reconhecida como função de avaliação, função de aptidão ou *fitness function*, é responsável por mensurar, através de métricas específicas, a qualidade de uma possível solução, possibilitando assim, a comparação e distinção sobre o que é um resultado bom ou ruim.

Consequentemente, os resultados tendem a ser superiores quando a etapa referente a modelagem do problema é bem definida, haja vista que o processo de busca será mais efetivo e, conseqüentemente, o tempo computacional será consideravelmente reduzido.

Com relação ao funcionamento genérico dos algoritmos populacionais de busca, o primeiro passo para o início da busca é verificar a presença de uma ou mais soluções no espaço de busca. Em seguida, após o estabelecimento de alguma condição de parada, os operadores de busca geram iterativamente novas soluções a partir das soluções atuais. Cada solução gerada é avaliada de acordo com a função objetivo, de tal maneira que as piores são substituídas por melhores durante o restante do processo. Ao fim, retorna-se a melhor solução encontrada durante a busca. Dentre as diversas técnicas de otimização baseadas em busca difundidas em SBSE, pode-se destacar os Algoritmos Evolutivos, Otimização por Colônia de Formigas, Subida da Colina e Têmpera Simulada (QU et al., 2009).

Portanto, o uso de algoritmos de busca demonstra-se deveras oportuno e relevante em problemas para os quais não existe algum método exato que retorne boas soluções em



tempo prático. Inclusive, Harman e Clark (2004) elencam algumas características necessárias que o problema a ser modelado deve apresentar como, por exemplo, grande espaço de busca, continuidade e baixa complexidade computacional na função objetivo e desconhecimento de soluções ótimas.

## 2.2 OTIMIZAÇÃO INTERATIVA

Apesar da intensa automatização promovida pela aplicação de SBSE, não se pode esquecer o quão crucial é a efetiva participação humana, afinal, é o usuário quem acumula conhecimento subjetivo e se capacita para melhor definir como serão os critérios qualitativos e quantitativos a serem utilizados no projeto. Assim, é fundamental destacar que o objetivo das metaheurísticas de busca é apoiar o ser humano, e não simplesmente substituí-lo.

Sistemas centrados em pessoas são pertinentes para problemas complexos e dinâmicos, pois conseguem proporcionar uma fusão entre o conhecimento tácito e a simulação computacional (PARMEE et al., 2006). Pelo protagonismo exercido nesse processo interativo, o tomador de decisão deve ter a responsabilidade de estar plenamente imerso nas peculiaridades do projeto, além de suficientemente consciente na avaliação dos dados e na introdução de informações externas.

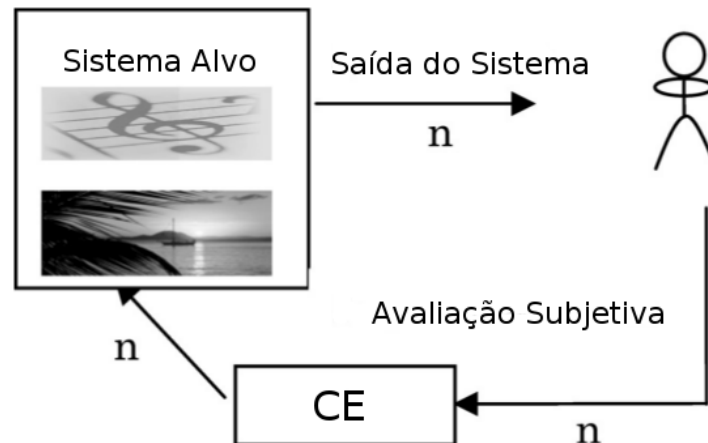
Conforme mencionado anteriormente, a Otimização Interativa mostra-se uma interessante abordagem de resolução para os problemas de SBSE devido a possibilidade de absorção do conhecimento humano na avaliação das soluções. Tal linha de pesquisa é bastante útil quando a função objetivo não pode ser explicitamente capturada ou em casos que o *know-how* do tomador de decisão pode contribuir efetivamente para resultados mais personalizados (TAKAGI, 2001).

Naturalmente, a Otimização Interativa também já foi explorada em diversos domínios de estudo, dentre os quais pode-se mencionar aplicações na área da computação gráfica (SIMS, 1991), robótica (LEWIS et al., 1992), reconhecimento facial (CALDWELL; JOHNSTON, 1991), criação de música (BILES, 1994), elaboração de vestuários (KIM; CHO, 2000) e geofísica (BOSCHETTI; MORESI, 2001).

Como uma ramificação da Otimização Interativa, surge a Computação Evolucionária Interativa (CEI), a qual é sustentada por dois componentes chaves, que são a avaliação humana e a busca computacional através de estratégias evolucionárias bioinspiradas (HARMAN, 2007). Wang et al. (2006) reitera tal conceito ao afirmar que CEI é uma tecnologia cooperativa que combina aspectos sensoriais, perceptivos e cognitivos do ser humano com a capacidade de busca

proveniente dos algoritmos oriundos da Computação Evolucionária (CE). De forma genérica, a Figura 3 mostra um sistema CEI através do qual o usuário vê ou ouve saídas, interage com CE, que conseqüentemente otimiza o sistema alvo para obter a saída preferida pelo usuário.

Figura 3 – Sistema CEI Genérico



Fonte: Adaptado de Takagi (2001).

Um interessante benefício obtido através da utilização da CEI que merece ser destacado, é a capacidade do ser humano adaptar sua avaliação subjetiva ao longo da evolução do algoritmo e, a partir das novas soluções apresentadas, obter *insights* que não haviam sido previstos.

A CEI tem um escopo bastante amplo que inclui diversas abordagens examinadas em trabalhos como (TAKAGI, 2001), onde são expostas várias aplicabilidades que podem ser conduzidas com base nesse princípio. De forma mais específica, em (HARMAN, 2007) é elucidada diversas oportunidades e desafios pertinentes da relação entre a CEI e a Engenharia de Software

Na CEI os usuários são convidados a interagir com o sistema durante o processo evolutivo e, conseqüentemente, são forçados a assumir um compromisso de resposta durante várias interações ao longo das gerações (HSU; HUANG, 2005). Uma grande quantidade de avaliações resulta no problema da fadiga humana, que por sua vez pode limitar o tamanho da população ou o número de gerações sobre as quais a evolução pode ocorrer, afetando diretamente o processo de busca (KAMALIAN et al., 2006). Alguns pesquisadores vêm constantemente propondo métodos que minimizem esse esforço, mas que ainda mantenham os resultados adequados.

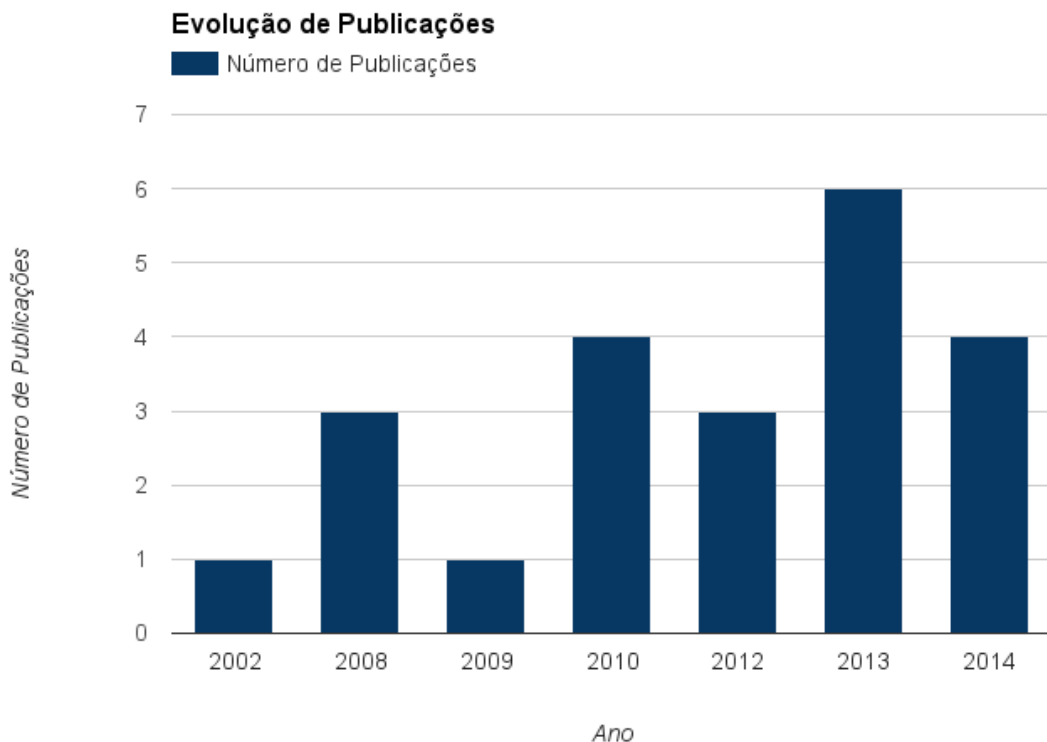
Dentre as técnicas de busca que podem ser aplicadas considerando a CEI, optou-se

no presente trabalho pela utilização dos Algoritmos Genéticos Iterativos devido a sua ampla aplicação em trabalhos que envolvem Otimização Iterativa e SBSE. A única diferença entre as duas versões reside no quesito referente a avaliação das soluções, haja vista que no IGA o ser humano é incluído de forma mais incisiva.

### 2.3 OTIMIZAÇÃO INTERATIVA EM ENGENHARIA DE SOFTWARE BASEADA EM BUSCA

Tendo em vista a constante aplicação dos conceitos de Otimização Iterativa em SBSE, neste tópico elabora-se uma breve análise bibliográfica sobre os trabalhos que exploram esta convergência. De maneira geral, esta aplicação pode ser analisada na Figura 4 onde viabiliza-se uma visão geral com relação ao número de publicações por ano.

Figura 4 – Número de publicações que utilizam conceitos de Otimização Iterativa em SBSE entre 2002 e 2014



Fonte: Adaptado de Zhang et al. (2011).

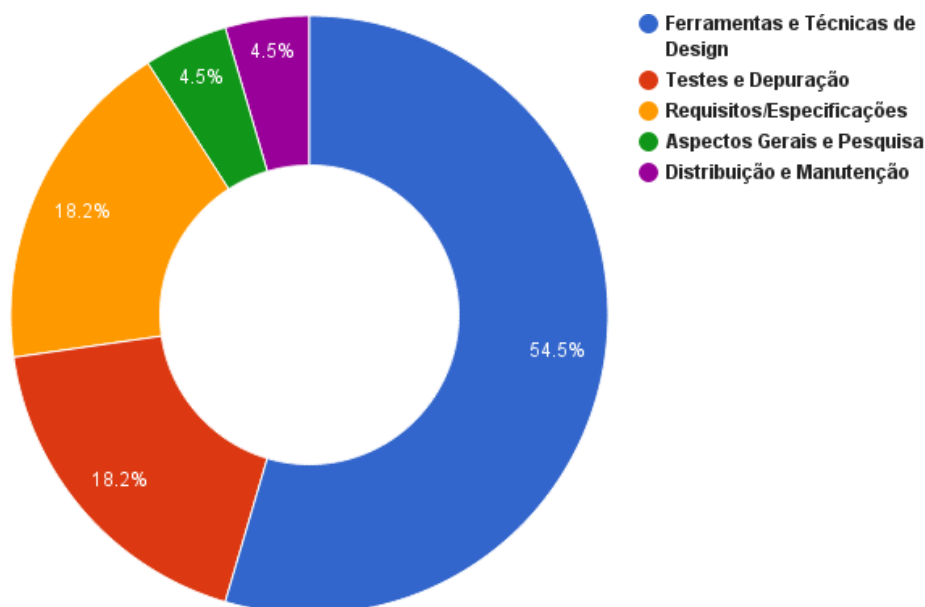
Em grande parte a motivação destes trabalhos derivam da contrariedade dos usuários em lidar resultados totalmente automáticos e que, por muitas vezes, não atendem a aspectos subjetivos difíceis de serem modelados matematicamente. Portanto, propõe-se a ideia de in-

corporar o conhecimento humano no processo de busca de modo que a geração das soluções seja devidamente influenciada. Assim, os resultados serão otimizados com a vantagem de que, mesmo perdendo um pouco em aspectos objetivos, os resultados estarão mais adequados em termos subjetivos.

Todavia, também é relevante ressaltar que a avaliação subjetiva humana tende a ser inconsistente e não-linear com o passar do tempo, necessitando de uma rápida convergência. Porém, ainda assim, considera-se bastante proeminente a estratégia de se manipular interativamente o processo de busca em prol de se viabilizar a conciliação entre os aspectos matemáticos e psicológicos (SIMONS et al., 2012).

Verifica-se na Figura 5 que diversos segmentos da SBSE já foram explorados. Com 54,5%, a área mais abordada é a de Ferramentas e Técnicas de Design. Tal valor pode ser justificado devido o diferencial em se apresentar aspectos visuais (diagramas de classe, por exemplo) e obter mais intuitivamente as avaliações subjetivas do usuário. Requisitos, por sua vez, vem logo em seguida como área mais explorada pela comunidade.

Figura 5 – Taxa de publicações que utilizam conceitos de Otimização Interativa em SBSE por área entre 2002 e 2014



Fonte: Adaptado de Zhang et al. (2011).

Apesar de prosseguir com os preceitos estabelecidos em SBSE tradicional, julga-se relevante que antes de propriamente aplicar os conceitos de Otimização Interativa sejam especificadas alguns fundamentos básicos que englobam duas etapas distintas de modelagem.

São elas:

- a) **Modelagem Matemática:** nesta etapa define-se toda a modelagem referente ao problema a ser explorado, como por exemplo, as representações das soluções, a função de avaliação, as restrições do problema e, claro, qual algoritmo de busca será utilizado.
- b) **Modelagem Interativa:** nesta fase objetiva-se decidir todos os aspectos que compõem a perspectiva interativa da abordagem, haja vista que existe uma latente necessidade de não sobrecarregar cognitivamente o usuário. Assim, existem algumas perguntas relevantes de serem discutidas, dentre as quais pode-se destacar:
- **Em qual momento a preferência será capturada?**  
Miettinen (1999) sugere três momentos diferentes de se incluir o tomador de decisão em métodos de busca interativa. Na opção *a priori*, o usuário deve especificar suas preferências antes de iniciar o processo de busca. Já no método *a posteriori*, a busca é inicializada sem a influência humana e só ao fim é disponibilizado um conjunto de soluções para que o tomador de decisão opte pela que melhor se adequa às suas opções. Por fim, é sugerida uma terceira possibilidade onde o usuário participa ativamente durante o processo de busca guiando as soluções de acordo com seus critérios particulares.
  - **Qual informação será concedida pelo usuário?**  
Neste quesito deve-se definir como o usuário irá expressar suas preferências em prol de um melhor direcionamento para o algoritmo. Consequentemente, verifica-se que diversas possibilidades tornam-se plausíveis, como por exemplo, fornecer uma avaliação numérica ou nominal para cada solução, distinguir subjetivamente um conjunto de soluções, influenciar como um critério de desempate, dentre outros.
  - **De que forma a opinião será incorporada e influenciará no processo de busca?**  
Conforme mencionado na primeira pergunta, a informação subjetiva poderá ser incluída em diversos momentos. Consequentemente, sua influência no processo de busca irá variar. Por exemplo, caso a opinião do usuário seja considerada como um objetivo na função de avaliação, presume-se que ao decorrer

do algoritmo todas soluções serão guiadas de acordo com seus critérios. Dessa forma, é natural refletir sobre o quão importante é a quantidade de interações a serem estabelecidas, dada a importância de se minimizar a fadiga humana.

Logicamente, por se tratar de uma abordagem em processo de consolidação, a aplicação de Otimização Interativa em SBSE enfrenta diversos desafios, dentre os quais pode-se mencionar:

- a) A busca por alternativas no tratamento da fadiga humana. Atualmente já se utilizam alguns recursos como uso de técnicas de inteligência artificial para predição de avaliações, agilizar a convergência do algoritmo ou proporcionar um espaço de busca mais completo.
- b) Adaptação de novos algoritmos de busca com viés interativo que possam ser aplicados em novas abordagens para problemas da Engenharia de Software.
- c) Definição de novas formas de interação que possibilitem ao usuário um menor desgaste cognitivo, mas que ainda sejam eficientes em termos de adequação subjetiva.
- d) Adequação em termos de usabilidade referentes a todas as interfaces que necessitam de alguma interação para um melhor rendimento humano.
- e) Amadurecimento na condição com que toda a experimentação humana será conduzida, cujo nível de configuração abrange diversas especificações discernidas na Engenharia de Software experimental.

## 2.4 ALGORITMO GENÉTICO INTERATIVO

Para um melhor entendimento do Algoritmo Genético Interativo, é necessário primeiramente compreender sua versão canônica, haja vista que seus princípios são reutilizados na abordagem interativa. Apesar de ter sido elaborado muito tempo antes, foi a partir das pesquisas desenvolvidas por John Holland em meados dos Anos 70 que o Algoritmo Genético em sua versão canônica começou a se consolidar. O grande diferencial dessa técnica consiste na possibilidade de resolver problemas de otimização e classificação através de conceitos oriundos da natureza (HOLLAND, 1975). De maneira geral, podem ser considerados como algoritmos probabilísticos que fornecem um mecanismo de busca paralela e adaptativa baseada na criação de uma população de indivíduos que será submetida a um processo evolutivo (PACHECO, 1999).

Tal algoritmo é fundamentado na concepção de que um “indivíduo” da população

representa uma solução candidata para dado problema. Posteriormente, cada indivíduo é imprerivelmente avaliado por intermédio de uma função objetivo, cuja responsabilidade consiste em auferir a aptidão (qualidade) da respectiva solução (LACERDA; CARVALHO, 1999). Os indivíduos da população são submetidos a um processo evolutivo no qual competem por recursos no meio de um ambiente, porém, são os melhores quem têm maior probabilidade de sobreviver e propagar o seu material genético para seus descendentes.

Através do princípio da *seleção* são escolhidos para o prosseguimento do algoritmo aqueles indivíduos da população cujo valor de aptidão é relativamente elevado. Em prol de uma evolução contínua será necessário submeter tais indivíduos aos operadores de *cruzamento* e *mutação*. Esses operadores são complementares e possuem papel fundamental por contribuírem no processo de geração de descendentes e numa melhor diversidade da população. Tal processo evolutivo referente à versão canônica do Algoritmo Genético pode ser visualizado no Algoritmo 1.

---

#### Algoritmo 1: Algoritmo Genético Canônico

---

**Saída:** Indivíduo com maior aptidão

**início**

Cria população inicial;

**enquanto** *não atingir o critério de parada* **faça**

    Calcula aptidão dos indivíduos da população;

**para** *cada indivíduo na população* **faça**

        Faz a seleção dos pais;

        Faz o cruzamento dos pais;

        Faz a mutação dos filhos;

**para** *cada filho inválido* **faça**

            Repara indivíduo;

**fim**

**fim**

    Atualiza população com novos filhos;

**fim**

**retorna** indivíduo com maior aptidão da população;

**fim**

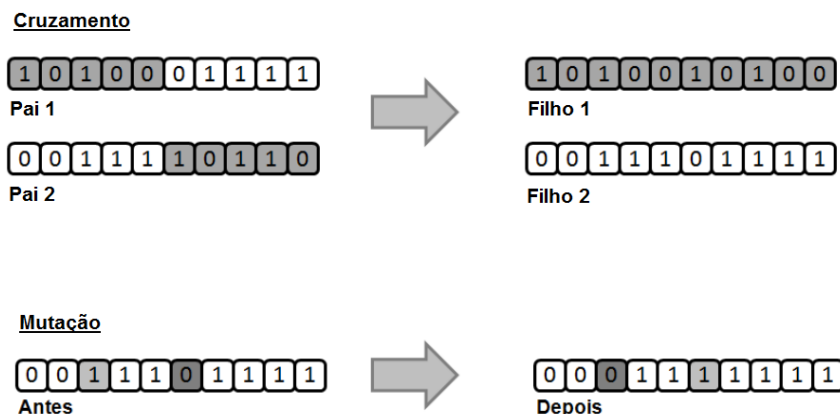
---

Existem diversos tipos de *cruzamento*, mas, de maneira geral, todos visam recombinar as características dos indivíduos durante a reprodução, proporcionando assim, que os futuros indivíduos herdem características de gerações anteriores. No presente trabalho foi utilizado o

“*cruzamento de um ponto*”, cujo funcionamento consiste em dividir os pais em cabeça e cauda a partir de um ponto de corte aleatoriamente definido. O primeiro filho é formado pela cabeça do primeiro pai em conjunto com a cauda do segundo, enquanto o segundo filho é constituído pela cabeça do segundo pai com a cauda do primeiro.

Já a *mutação* tem a responsabilidade de modificar o valor de um ou mais genes do indivíduo para garantir uma maior varredura do espaço de estados e assim evitar que o algoritmo convirja muito cedo para mínimos locais. Dentre as diversas estratégias de mutação, optou-se pela versão mais difundida denominada como *bit flip*. Nessa abordagem ocorre um sorteio aleatório com o desígnio de alterar um valor de um gene específico de cada indivíduo, portanto, pode-se interpretar simplesmente como uma inversão do valor de um dado bit de 1 para 0 ou de 0 para 1, caso a representação seja binária. Na Figura 6 são ilustrados dois exemplos que elucidam respectivamente a utilização dos operadores de cruzamento e mutação.

Figura 6 – Exemplos de Cruzamento e Mutação



Fonte: Maia (2011).

Naturalmente, após a ação dos operadores de cruzamento e mutação, podem ser gerados indivíduos inválidos. Entende-se como indivíduo inválido aquele que não atende todas restrições modeladas no problema. Assim, ressalta-se a presença de um outro operador denominado *reparação*, cujo objetivo consiste justamente em transformar um indivíduo inválido num válido. Outro aspecto importante a se mencionar é a presença do princípio do *elitismo*, o qual visa evitar que possíveis bons indivíduos se percam durante a evolução do algoritmo. Tal estratégia objetiva fundamentalmente copiar uma certa porcentagem dos melhores indivíduos da população corrente para à próxima geração, garantindo assim, que o melhor indivíduo referente à próxima geração não será pior que um da anterior.



Todavia, falta às aplicações que envolvem Algoritmos Genéticos a capacidade de utilizar a opinião humana como meio efetivo de apoio a tomada de decisão. Logo, surge a oportunidade de se utilizar como alternativa para este impasse sua abordagem interativa, ou seja, o Algoritmo Genético Interativo. Tal abordagem segue os mesmos conceitos de sua versão canônica, exceto no quesito referente à função objetivo, afinal como o mesmo segue as diretrizes da Computação Evolucionária Interativa. Isto é, será o usuário quem terá a responsabilidade de influenciar no cálculo da aptidão para cada indivíduo ao invés de uma função convencional (KIM; CHO, 2000).

Neste caso, é pertinente enfatizar que a função objetivo pode tanto ser substituída completamente, quanto apenas influenciada de forma parcial. Com isso, torna-se plenamente possível proporcionar ao algoritmo uma maior interatividade com o usuário, de modo que suas preferências serão devidamente incorporadas ao longo do processo evolutivo.

Portanto, pode-se concluir que o princípio básico de um Algoritmo Genético Interativo consiste em transformar a população através de sucessivas gerações com o suporte dos operadores genéticos e da opinião humana. Tal ciclo é estendido até um número de vezes previamente estabelecido ou até chegar a um resultado considerado satisfatório. Assim, considera-se relevante a possibilidade de proporcionar que as soluções sejam constantemente complementadas pelo conhecimento humano.

## 2.5 APRENDIZADO DE MÁQUINA

Aprendizado de Máquina é uma área da Inteligência Artificial que visa o desenvolvimento de processos de aprendizado e construção de sistemas aptos a automaticamente adquirir conhecimento e novas habilidades. Um sistema de aprendizado consiste num programa de computador cuja tomada de decisão é efetivada a partir de um conjunto de experiências acumuladas previamente, ou seja, o desempenho de determinada tarefa é equivalente ao seu grau de experiência aprendido (MITCHELL, 1997). Tal área é reconhecidamente multidisciplinar e engloba métodos que conseguem colaborar com a automação na aquisição, estruturação, acesso do conhecimento adquirido e, conseqüentemente, promova um comportamento inteligente (MONARD; BARANAUSKAS, 2003).

Portanto, Aprendizado de Máquina é a programação de computadores com o objetivo de otimizar um critério de desempenho utilizando dados de exemplo ou experiências passadas, sendo intrinsecamente relacionado a identificação de padrões ou regularidades. Após estabelecer

o modelo e seus parâmetros, o aprendizado é justamente a execução de um programa que visa otimizar tais parâmetros do modelo a partir de dados de treinamento ou experiências anteriores. O modelo pode ser de natureza preditiva (visa prever o futuro), descritiva (objetiva ganhar conhecimento dos dados) ou ambos os tipos (ALPAYDIN, 2014). Através da Figura 7 é possível verificar a classificação dos sistemas de Aprendizado de Máquina, cujos Witten e Frank (2005) definem como:

a) **Aprendizado Supervisionado:** neste tipo, o técnica de aprendizado recebe um conjunto de exemplos de treinamento, onde cada exemplo é associado a um rótulo conhecido. Tal rótulo é responsável por definir a qual classe o respectivo exemplo (instância) pertence. Cada exemplo é descrito como um vetor de valores com as características (atributos), além do seu respectivo rótulo. Nesse tipo, a técnica de aprendizado é capaz de criar uma classificação a partir de um conjunto de exemplos, onde o principal objetivo consiste em induzir um mapeamento geral dos vetores de características para os valores das classes, de tal maneira que se permita ao sistema de aprendizado construir um modelo de uma função desconhecida que seja apta a prever rótulos de classes para exemplos previamente não vistos. Em casos nos quais os valores do rótulo da classe podem ser qualitativos e assumir um conjunto de valores discretos, a tarefa é nomeada de *classificação*. Se os valores dos rótulos forem de natureza quantitativa, assumem um conjunto de valores contínuos e a tarefa é chamada de *regressão*.

b) **Aprendizado Não-Supervisionado:** neste caso é fornecido ao sistema de aprendizado um conjunto de exemplos de maneira que o objetivo seja construir um modelo que procure regularidades em tais exemplos, formando assim, agrupamentos ou *clusters* de características similares. Assim, após formados esses agrupamentos determina-se o que cada um significa sob o contexto do problema.

Na literatura é possível constatar que existem diversos paradigmas de aprendizado automático capazes de aprender a partir de um conjunto de exemplos, são eles:

a) **Simbólico:** o processo de aprendizado é construído através da construção de representações simbólicas de um determinado conceito através de exemplos e contra-exemplos do mesmo. As representações simbólicas estão tipicamente na forma de alguma expressão lógica, como por exemplos, árvores de decisão,

Figura 7 – Classificação dos Sistemas Aprendizado de Máquina



Fonte: Carvalho (2002).

regras de decisão ou redes semânticas (MONARD; BARANAUSKAS, 2003).

- b) **Estatístico:** como regra geral, técnicas estatísticas tendem a focar tarefas em que todos os atributos têm valores contínuos ou ordinais. Neste paradigma objetiva-se elaborar um modelo estatístico que se aproxime o máximo possível do conceito induzido. A maioria destes modelos se caracterizam por serem paramétricos, ou seja, os parâmetros utilizados são ajustados visando uma melhor aproximação sobre o conjunto de dados. Dentre os métodos mais utilizados pode-se destacar o Aprendizado Bayesiano (REZENDE, 2003).
- c) **Baseado em Exemplos:** tal paradigma é fundamentado no processo de comparação das características de um exemplo a ser classificado com apoio de uma base de exemplos classificados. Assim, presume-se que a classe do exemplo é a mesma classe daqueles exemplos previamente conhecidos que também possuem as mesmas características. Alguns dos algoritmos mais importantes deste paradigma são o Raciocínio Baseado em Casos e o *Nearest Neighbours*.
- d) **Conexionista:** baseia-se nas inter-conexões entre os atributos representativos de uma classe. O nome *conexionismo* é utilizado para descrever a área de estudo que foca nas construções matemáticas inspiradas no modelo biológico do sistema nervoso. Redes Neurais é a técnica de maior destaque desse paradigma (REZENDE, 2003).
- e) **Genético:** inspira-se na seleção natural proposta por Charles Darwin. O paradigma genético utiliza a evolução de uma população de classificadores que

competem para fazer a predição, sendo selecionado aquele que obter os melhores resultados. Dentre as técnicas exploradas pode-se citar a Programação Genética e os Algoritmos Genéticos (KOZA, 1992).

### 2.5.1 Redes Neurais

Sob a ótica dos paradigmas de Aprendizado de Máquina, existem as técnicas ou algoritmos que são responsáveis pela forma de aprendizado. Dentre as técnicas mais utilizadas pode-se destacar as Redes Neurais Artificiais (RNA) devido, entre outros motivos, ao seu alto grau de empregabilidade.

O conceito de RNA's inspira-se na simulação de comportamentos inteligentes através de modelos baseados na estrutura e funcionamento do cérebro humano. De maneira geral, podem ser concebidas como sistemas paralelos distribuídos constituídos por unidades de processamento simples (neurônios) que computam determinadas funções matemáticas.

Esses neurônios podem estar dispostas em uma ou mais camadas. Existem três tipos de camadas: *entrada* cujo objetivo consiste receber os sinais de entrada; *saída* da qual se extrai os sinais de saída e *intermediárias* que podem ou não existir. Essas camadas são interligadas por um grande número de conexões, que por sua vez, estão associadas a pesos que objetivam ponderar as entradas recebidas por cada neurônio e armazenar o conhecimento representado no modelo.

As RNA's costumam ser classificadas de acordo com sua arquitetura, ou seja, o modo como seus neurônios são estruturados e organizados. De acordo com Akerkar e Sajja (2010), existem quatro tipos:

- a) *Single layer feed-forward network*: possui apenas uma camada de entrada e outra de saída. É denominada *feed-forward* porque a alimentação da rede propaga-se apenas progressivamente.
- b) *Multilayer feed-forward network*: é uma rede *feed-forward* que, além das camadas de entrada e saída, conta com uma ou mais camadas escondidas. Se todos neurônios de cada camada da rede é conectada a sua camada adjacente, a rede é chamada de totalmente conectada. Já se alguma dessas conexões não são concluídas, a rede é dita parcialmente conectada.
- c) *Recurrent neural networks*: rede cuja saída de um neurônio pode ser entrada para outro de uma camada precedente. Também há o caso no qual ocorre a

auto-realimentação, ou seja, a saída de um neurônio pode ser entrada para ele mesmo.

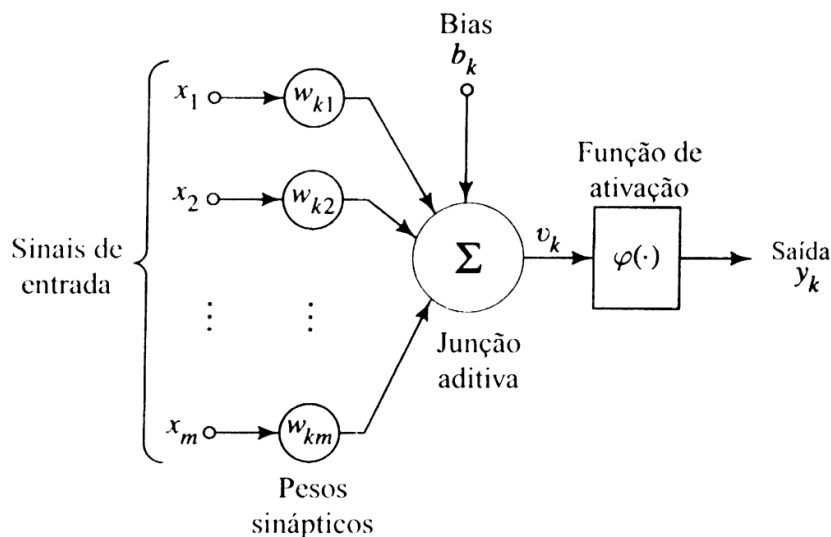
- d) *Lattice networks*: rede constituída de neurônios de saída organizados em linhas e colunas. Assim, a mesma pode ter diversas dimensões de neurônios que fornecem sinais de entrada.

Dessa forma, pode-se verificar que um dos principais benefícios oriundos da aplicação de uma rede neural é o aprendizado, no qual é alcançado justamente pelo mapeamento de entrada e saída que são fornecidos aos pesos sinápticos para adequar devidamente as saídas às entradas.

O que diferencia uma RNA é a configuração do seu processo de treinamento. Na fase de treinamento a rede neural ajusta seus pesos através do uso de um algoritmo apropriado, onde posteriormente compara-se com a saída alcançada. Tal diferença é incorporada no ajuste dos pesos sinápticos. Logo após, na fase de validação, a RNA calcula a saída com base nas entradas e nos pesos.

Na Figura 8 é representado um modelo de um neurônio.  $X_j$  são as entradas do neurônio  $K$ ,  $W_{kj}$  são os pesos sinápticos do neurônio  $K$  de cada entrada  $j$ , e  $b_k$  tem a função de aumentar ou diminuir a entrada líquida da função de ativação mostrada na Equação 2.1.

Figura 8 – Modelo de um Neurônio



Fonte: Haykin (2001).

$$U_j = \sum_{j=1}^m W_{kj} X_j \text{ e } V_k = U_k + b_k \text{ e } Y_k = \phi(U_k + b_k) \quad (2.1)$$

onde,  $X_j$  refere-se ao sinal de entrada,  $W_{kj}$  aos pesos sinápticos do neurônio,  $U_j$  a saída do combinado linear,  $V_k$  o campo local induzido e, por fim, o *bias*  $b_k$  o qual serve como peso extra.

Dependendo do tipo de função de ativação, o neurônio pode ser determinístico ou estocástico. Em um modelo genérico de um neurônio, a função de ativação é quem determina a atividade do mesmo, ou seja, processa o sinal gerado pela combinação linear das entradas e dos pesos das sinapses e, conseqüentemente, gera-se o sinal de saída do neurônio. Assim, a mesma fundamentalmente define a saída de um neurônio em termos do potencial de ativação  $v$ . No neurônio determinístico, a função de ativação (representada por  $\phi(V)$ ) pode ser de três tipos. Suas definições matemáticas e respectivas representações gráficas são mostradas a seguir:

- a) **Limiar:** representa uma função de ativação abrupta. Descreve a propriedade tudo-ou-nada, logo, a saída do neurônio é igual a zero, quando seu valor for negativo e 1, quando seu valor for positivo.

$$\phi(V) = \begin{cases} 1 & \text{se } V \geq 0 \\ 0 & \text{se } V < 0 \end{cases} \quad (2.2)$$

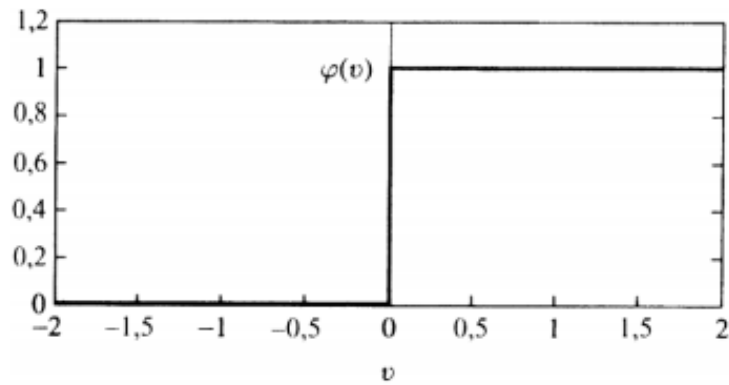
- b) **Limiar por partes:** resulta num combinador linear se a operação for restrita à região linear. Assim, funciona como uma aproximação de um amplificador não linear.

$$\phi(V) = \begin{cases} 1 & \text{se } V \geq \frac{1}{2} \\ V & \text{se } -\frac{1}{2} < V < \frac{1}{2} \\ 0 & \text{se } V \leq -\frac{1}{2} \end{cases} \quad (2.3)$$

- c) **Sigmóide:** função crescente que apresenta um balanceamento entre o comportamento linear e não-linear. Exemplos dessa função são a função logística e a tangente hiperbólica.

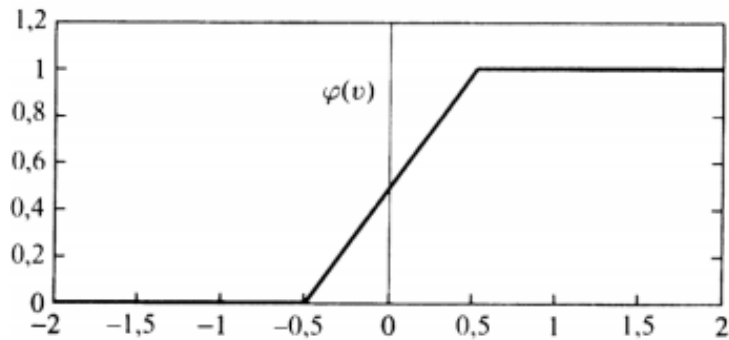
$$\phi(V) = \frac{1}{1 + e^{-aV}} \quad (2.4)$$

Figura 9 – Gráfico da função Limiar



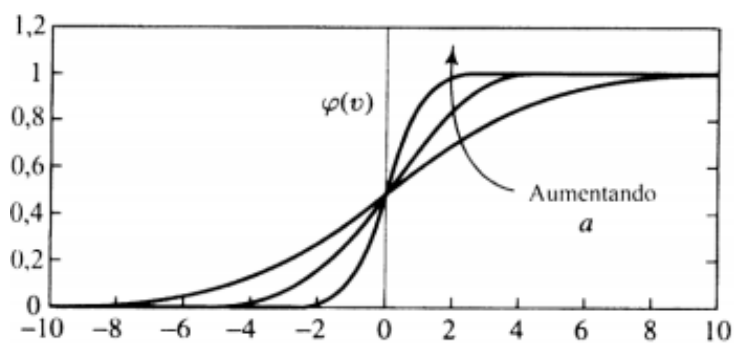
Fonte: Haykin (2001).

Figura 10 – Gráfico da função Limiar por partes



Fonte: Haykin (2001).

Figura 11 – Gráfico da função Sigmóide



Fonte: Haykin (2001).

Portanto, pode-se perceber que uma rede neural é caracterizada por diversos fatores como, a topologia da rede, o valor do peso da conexão entre os pares de neurônios e pelas regras de atualização de estado. Assim, constata-se que a etapa a qual envolve a definição arquitetural de uma RNA é muito importante.

## 2.5.2 Técnicas Aplicadas

O problema explorado no presente trabalho envolve aproximação funcional, a qual consiste em mapear o comportamento de um processo baseando-se em diversas medições efetivadas em suas entradas e saídas (SILVA et al., 2010). Dessa forma, aplica-se o conceito de aprendizado a partir de exemplos, onde é reproduzido o comportamento do sistema a partir da tradução de um conjunto de entradas e saídas.

No presente trabalho optou-se pela utilização do *Least Median Square* e *Multilayer Perceptron* por, além de serem amplamente utilizados, proporcionarem uma análise distinta dada as particularidades que cada técnica possui.

Logo, julga-se proeminente a aplicação de técnicas como LMS e MLP quando se possui previamente uma coleção de dados de entradas e saída, como é o caso desta pesquisa. Dessa forma, deseja-se demonstrar a utilização dos conceitos de aprendizado de máquina na aproximação do comportamento funcional do processo.

### 2.5.2.1 Least Median Square

O *Least Median Square* (LMS), também conhecido como Regra Delta, foi precursor do primeiro algoritmo para treinamento de redes perceptron de múltiplas camadas: o *backpropagation* (BRAGA et al., 2000). Criado por Widrow e Hoff em 1960, é bastante utilizado devido sua natural capacidade de resolver problemas lineares e não-lineares.

O LMS utiliza dos conceitos da Regressão Linear Simples para formar previsões (ROUSSEEUW; LEROY, 2005). Dito isso, a regressão visa a busca por uma função que consiga se aproximar do comportamento definido pelo caso estudado. A abordagem mais difundida é a linear o qual pode ser considerado como um método que minimiza o erro médio entre os valores esperados de uma variável  $y$ , dado os valores outras variáveis  $x$ .

Considerando a disponibilidade de diversas amostras de treinamento a ideia fundamental na aplicação do LMS envolve ajustar o vetor de pesos  $e$ , conseqüentemente, minimizar a diferença entre a saída desejada e a resposta do combinador linear (SILVA et al., 2010) descrita na Equação 2.5:

$$J = \frac{1}{2} \sum_{i=1}^p (\gamma_d^i - (w \cdot x^i))^2 \quad (2.5)$$



Dessa forma, para uma condição inicial de pesos qualquer  $\mathbf{w}^{(0)}$ , objetiva-se obter a direção do ajuste a ser aplicado no vetor de pesos de forma tal que se aproxime do mínimo da Equação 2.5. Para sua superfície de erro, a direção do ajuste na iteração  $n$  pode ser obtida através do seu gradiente no ponto  $\mathbf{w}^{(n)}$  (BRAGA et al., 2000). Como o vetor gradiente possui a mesma direção da maior variação do erro, o ajuste dos pesos deve ocorrer em direção contrária a este, ou seja,  $\Delta\mathbf{w}^{(n)} \propto -\nabla J$ . As derivadas parciais em relação a cada um dos pesos podem ser obtidas através das Equações 2.6, 2.7 e 2.8:

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial \gamma} \frac{\partial \gamma}{\partial w_i} \quad (2.6)$$

$$\frac{\partial J}{\partial w_i} = -x_i(d - (w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n)) \quad (2.7)$$

$$\frac{\partial J}{\partial w_i} = -x_i e \quad (2.8)$$

Logo, para que se obtenha o valor dos pesos ajustado em direção contrária ao vetor gradiente chega-se às seguintes equações de ajuste:

$$\Delta \mathbf{x}_i \propto e x_i \quad (2.9)$$

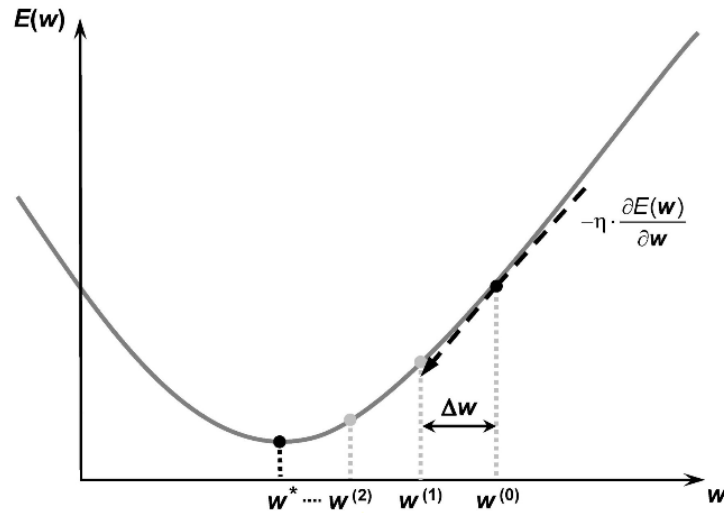
$$\Delta \mathbf{x} = \eta e x_i \quad (2.10)$$

onde  $\eta$ , também reconhecido como taxa de aprendizado, exprime quão rápido o treinamento estará rumando em direção ao ponto de minimização da Equação 2.5. Normalmente adota-se valores pertencentes ao intervalo compreendido em  $0 < \eta < 1$  (SILVA et al., 2010). Finalmente, a equação genérica para ajuste dos pesos pode ser obtida como:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta e \mathbf{x}(n) \quad (2.11)$$

Através da Figura 12 demonstra-se o processo de convergência do algoritmo contendo uma interpretação geométrica referente aos passos de atualização do vetor de pesos rumo ao ponto de minimização da Equação 2.5.

Figura 12 – Interpretação Geométrica do *Least Median Square*



Fonte: Silva et al. (2010).

Assim, é possível constatar que partindo de um valor inicial ( $w^{(0)}$ ), o próximo valor de  $w$  (representado por  $w^{(1)}$ ) será obtido considerando-se a direção oposta ao vetor gradiente em relação ao valor de  $w^{(0)}$ . Para o próximo passo de atualização o ajuste do peso  $w$  (agora representado por  $w^{(2)}$ ), será realizado considerando o valor do gradiente em relação a  $w^{(1)}$ . Conseqüentemente, aplicando tais passos sucessivamente, o processo de convergência estará rumando de forma iterativa em direção ao valor de  $w^*$ , sendo este a configuração ótima. Após o processo ter convergido para  $w^*$ , o valor de  $E(w)^*$ , será então sempre menor que quaisquer  $E(w)$  calculados nos passos anteriores.

Portanto, em síntese o LMS busca convergir o vetor de pesos até um valor aproximado, cuja função tem uma determinada taxa de variação máxima. Esse processo utiliza-se de um valor de gradiente objetivando diminuir o erro.

### 2.5.2.2 Multilayer Perceptron

No início da década de 70 apresentou-se o conceito das Redes Perceptron de Múltiplas Camadas, em inglês *Multilayer Perceptron* (MLP), cuja característica marcante consiste na composição de uma camada de entrada, uma ou mais camadas intermediárias e uma de saída. Essa visão amplia o número de problemas capazes de serem resolvidos por tais técnicas. Essa

proposta veio em resposta às limitações do modelo de rede neural Perceptron, que devido a sua conexão simples de apenas um neurônio mostra-se capaz de solucionar somente problemas linearmente separáveis. Barreto (1999) ressalta que uma rede MLP apresenta três características principais:

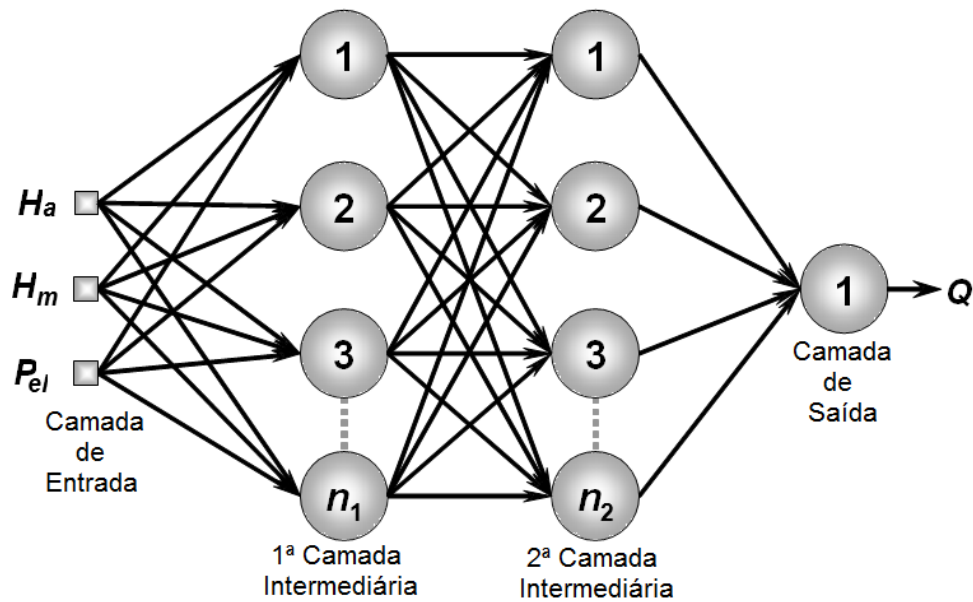
- a) O modelo de cada neurônio ou elemento processador da rede possui uma função de ativação não-linear. A função sigmoideal atende esta exigência e é muito utilizada em redes MLP.
- b) Possui pelo menos uma camada intermediária de processamento com neurônio que não faz parte da entrada ou saída.
- c) Possui alto grau de conectividade entre seus elementos processadores, a qual é definida através dos pesos sinápticos.

As redes MLP são reconhecidas pelo alto grau de aplicabilidade em diversos problemas das mais diferentes áreas do conhecimento, sendo considerada uma abordagem deveras versátil. Entre as potenciais áreas, destaca-se:

- a) Aproximação universal de funções;
- b) Reconhecimento de padrões;
- c) Identificação e controle de processos;
- d) Previsão de séries temporais;
- e) Otimização de sistemas.

Seguindo a classificação arquitetural exposta anteriormente, a rede MLP pertence à arquitetura *Multilayer feed-forward network*. Seu processo de treinamento é baseado no aprendizado supervisionado, isto é, para cada amostra dos dados de entrada obtém-se a respectiva saída desejada. Conforme pode-se observar na Figura 13, o fluxo de informações se inicia na camada de entrada, percorre as camadas intermediárias e, ao fim, finaliza-se na camada neural da saída.

Em termos de funcionamento, a ideia fundamental é que os sinais sejam apresentados à rede em sua camada de entrada. As camadas intermediárias extraem a maioria das informações referente ao comportamento e as codificam por meio dos pesos sinápticos e limiares de seus neurônios, formando assim uma representação própria do ambiente no qual está inserido o referido sistema. Por fim, os neurônios da camada de saída recebem os estímulos advindos da última camada intermediária, produzindo um padrão de resposta que será a saída disponibilizada pela rede (SILVA et al., 2010). Durante o processo de treinamento, cada apresentação completa

Figura 13 – Modelo *Multilayer Perceptron*

Fonte: Adaptado de Haykin (2001).

das amostras ao subconjunto de treinamento, visando o ajuste dos pesos, é denominado como época de treinamento.

Verifica-se que, cada uma das entradas da rede será propagada uma a uma em direção à camada neural de saída. Assim, independentemente do número de camadas intermediárias, a propagação dos sinais de entradas é sempre realizada num único sentido, ou seja, da camada de entrada em direção à camada de saída.

O algoritmo aplicado no decorrer do processo de treinamento é o *backpropagation*, comumente reconhecido como regra Delta generalizada. Esse processo de treinamento é realizado em duas fases específicas:

- a) Fase *Forward*: os sinais de uma amostra do conjunto de treinamento são inseridos nas entradas e propagados camada a camada até a produção das respectivas saídas. Assim, para obter as resposta da rede, leva-se em consideração apenas os valores atuais de pesos sinápticos e limiares de seus neurônios, os quais permanecerão inalterados durante toda esta primeira fase. Em seguida, as respostas produzidas são comparadas com as respectivas respostas almejadas que estejam disponíveis. Por fim, calcula-se os erros entre as respostas desejadas e aquelas produzidas pelos neurônios de saída.
- b) Fase *Backward*: considerando os valores de erros entre as respostas do estágio anterior, efetua-se os ajustes dos pesos sinápticos e limiares de todos os neurônios

da rede no decorrer desta fase.

Em síntese, as aplicações sucessivas das fases *forward* e *backward* fazem com que os pesos sinápticos e limiares dos neurônios se ajustem automaticamente em cada iteração, implicando-se na gradativa diminuição da soma dos erros produzidos pelas respostas da rede frente às desejadas.

### 2.5.3 Ferramenta WEKA

O *Waikato Environment for Knowledge Analysis* (WEKA) (HALL et al., 2009) é um produto *open source* da Universidade de Waikato (Nova Zelândia) que auxilia na utilização de técnicas de aprendizado de máquina. O mesmo se caracteriza por possuir um alto grau de portabilidade e modificabilidade. Em termos de utilização, viabiliza-se sua aplicação através de uma interface gráfica interativa ou também através de uma série de *Application Program Interface* (API's) gerais que facilitam a sua incorporação em outros projetos. Esta ferramenta implementa um grande conjunto de *features* que contribuem em atividades como, preparação dos dados e verificação/validação dos resultados.

Desde que foi disponibilizado no *Source Forge* em abril de 2000, o *WEKA* vem ampliando ainda mais sua aceitação nos meios acadêmicos e empresariais. Sua comunidade é bastante ativa e o mesmo já foi baixado mais de 1 milhão de vezes. (HALL et al., 2009). Tal ferramenta se destaca por contemplar diversas técnicas de aprendizado de máquina (supervisionado ou não), incluindo as utilizadas no presente trabalho.

## 2.6 PROBLEMA DO PRÓXIMO RELEASE

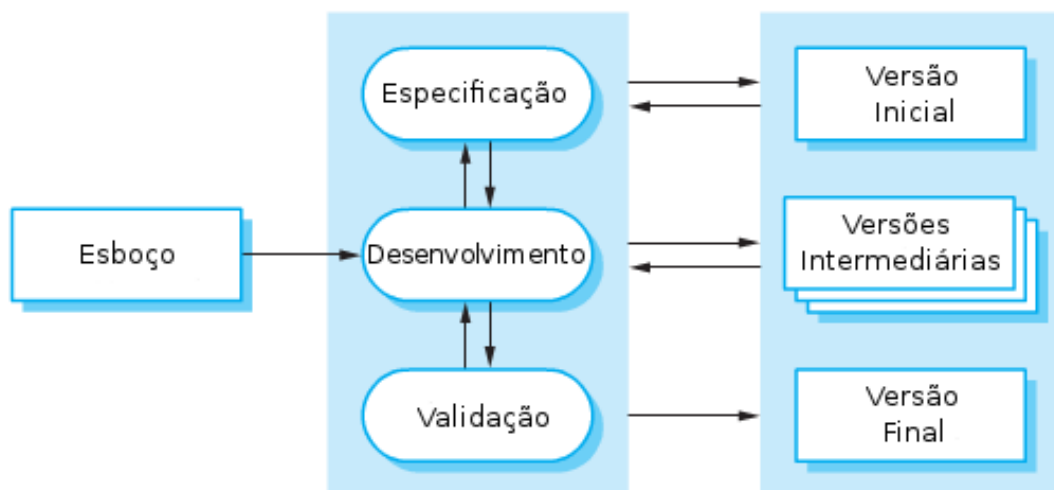
A partir dos serviços que se pretende oferecer, um sistema deve ter bem definido quais serão suas restrições e funcionalidades a serem desempenhadas. As descrições de tais necessidades são conhecidas como “requisitos”, cuja grande responsabilidade consiste em refletir as necessidades dos clientes (PRESSMAN, 2006). Dito isso, pode-se constatar o quão importante e expressiva é a seleção de requisitos no desenvolvimento de um software, afinal, lida diretamente com diversos fatores significativos para o resultado final do projeto, como aspectos financeiros e as próprias expectativas dos *stakeholders*.

Sabe-se que é muito importante que as organizações sejam capazes de lidar com as mudanças e suas consequências no processo de desenvolvimento de software, pois, é natural

que certas atividades sejam repetidas em respostas às inúmeras alterações que indubitavelmente ocorrerão durante a evolução do sistema. Mudanças nos requisitos requerem retrabalho do projeto e da implementação. Dessa forma, é fundamental estar preparado para elas (SOMMERVILLE, 2011).

Para suprir essas necessidades e as limitações de outros processos de desenvolvimento, como por exemplo o método cascata, surge o ciclo de vida iterativo e incremental. Um aspecto a se destacar de tal processo diz respeito à distinção das etapas de especificação, projeto e implementação de software em uma série de incrementos que são desenvolvidos gradativamente (SOMMERVILLE, 2011). Tais incrementos são denominados como *releases*, ou seja, uma versão intermediária do produto que seja devidamente executável e estável. Após a conclusão e entrega de um determinado *release*, o cliente tem a possibilidade de obter com antecedência e utilizar uma ou mais funcionalidades do sistema. Conforme pode-se visualizar na Figura 14, estes ciclos de incrementos (*releases*) se repetem iterativamente até que o sistema seja plenamente finalizado.

Figura 14 – Método Iterativo e Incremental



Fonte: Adaptado de Sommerville (2011).

Em um processo de desenvolvimento iterativo e incremental, a fase de especificação é elaborada em conjunto com o desenvolvimento do software, de maneira que o cliente coopera e atua incisivamente na definição de quais serviços o sistema deve prover, possibilitando assim, uma maior agilidade e adequação às suas preferências. Sommerville (2011) elenca uma série de vantagens que tal procedimento possibilita:

- a) Os clientes não precisam aguardar até a entrega do sistema inteiro para se

beneficiarem.

- b) Os clientes podem usar os incrementos iniciais como protótipos e, assim, ganhar experiência ao emitir os próximos *feedbacks* para a equipe.
- c) Maior satisfação do cliente com relação a prazo, haja vista que uma versão do sistema é entregue ao final de cada ciclo de entrega.
- d) Os serviços de prioridade mais alta são entregues primeiro.

Durante o desenvolvimento de um sistema existirão diversos *releases* a serem entregues, logo, a decisão sobre quais requisitos devem ou não ser incluídos para cada novo ciclo de entrega é muito relevante para a evolução do projeto. No presente trabalho almeja-se solucionar o Problema do Próximo *Release*, através de uma adaptação da modelagem proposta por Baker et al. (2006), o qual visa selecionar o melhor conjunto de requisitos a serem entregues no próximo *release*.

Nessa abordagem podem haver dois cenários: o primeiro, no qual o cliente é uma única pessoa e o segundo no qual existe uma organização que contém vários clientes (departamentos, por exemplo) com diferentes prioridades e níveis de importância. A ideia utilizada para contornar essas questões é o estabelecimento de uma certa importância para cada possível cliente. Assim, seja  $R = \{r_1, r_2, r_3, \dots, r_N\}$  o conjunto de todos os requisitos que deverão ser implementados no próximo *release* do sistema e  $C = \{c_1, c_2, c_3, \dots, c_M\}$  o conjunto de clientes atendidos pelo sistema, cada cliente  $c_j$  terá um determinado peso  $w_j \in Z^+$  que mensura sua respectiva importância para a empresa.

Em seguida, necessita-se que cada cliente tenha a responsabilidade de atribuir uma certa importância específica para cada requisito. Assim, seja  $S$  uma matriz de  $M \times N$  valores que representam a importância que um cliente  $m$  quantifica para cada requisito  $n$  em específico:

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{M,1} & s_{M,2} & \cdots & s_{M,N} \end{pmatrix} \quad (2.12)$$

Dada a importância específica ( $s_{ji}$ ) de um requisito  $r_i$  e o peso ( $w_j$ ) referente a cada cliente  $c_j$ , a importância total ( $v_i$ ) de se selecionar um requisito é dada pelo somatório dos produtos dos pesos de cada cliente ( $w_j$ ) pela importância específica ( $s_{ji}$ ) que cada um fornece ao requisito  $r_i$  (Equação 2.13). Portanto, sabendo que  $R' \subseteq R$  é um subconjunto de requisitos

selecionados para o próximo *release*, para se obter o seu valor de *score* calcula-se o somatório referente a importância total ( $v_i$ ) de cada requisito incluso na solução (Equação 2.14).

$$v_i = \sum_{j=1}^M w_j \times s_{ji} \quad (2.13)$$

$$score(R') = \sum_{v \in R'} v_i \quad (2.14)$$

Em prol de uma representação mais real para a proposta, cada requisito  $r_i$  ainda possui um custo de desenvolvimento  $e_i \in Z^+$ . Consequentemente, cada *release* está sujeita a uma restrição de *budget*, que no caso se refere ao orçamento disponível para o projeto. Logo, ao selecionar os requisitos para um determinado *release* deve-se atentar para seu respectivo custo total, haja vista que o somatório dos custos individuais ( $e_i$ ) relativos aos requisitos selecionados não podem exceder o *budget* previamente definido (Equação 2.15).

$$cost(R') = \sum_{i \in R'} e_i \quad (2.15)$$

Dessa forma, o NRP objetiva, através da seleção de um sub-conjunto de requisitos maximizar a importância total dos requisitos inclusos no próximo *release* de tal maneira que se respeite um certo orçamento, onde quanto mais requisitos estipulados como importantes pelos clientes são incluídos no *release*, maior será seu *score* (Equação 2.16).

$$\begin{aligned} &\text{maximizar } score(R') \\ &\text{sujeito a } cost(R') \leq budget \end{aligned} \quad (2.16)$$

Assim, fica notório que um dos principais benefícios promovidos pela SBSE é justamente a automatização proporcionada pela otimização no processo de resolução dos problemas. Entretanto, ao desconsiderar a participação efetiva humana em tal tomada de decisão, diversos benefícios que poderiam ser incorporados aos resultados são desconsiderados. Outro ponto significativo a se considerar, é que o nível de confiança dos usuários quando se deparam com resultados puramente automáticos tende a ser baixo, o que pode acarretar em uma série de dificuldades no progresso do projeto.



Portanto, entende-se que incorporar o ser humano durante o processo de busca demonstra-se uma abordagem relevante para resolução do NRP, inclusive, aproveitando o fato de que muitas organizações realizam o planejamento do próximo *release* de forma *ad hoc* ou artesanal (RUHE; SALIU, 2005). Dessa forma, as soluções continuarão a usufruir dos benefícios oriundos da otimização matemática, mas com a diferença de que as mesmas serão guiadas de acordo com as preferências e a subjetividade humana.

## 2.7 CONCLUSÕES DO CAPÍTULO

O presente capítulo objetivou proporcionar uma fundamentação teórica sobre os temas explorados neste trabalho. Acredita-se que tais discussões irão propiciar uma melhor compreensão das propostas que serão apresentadas adiante. Inicialmente, foram expostas as características que envolvem a Engenharia de Software Baseada em Busca, cujo propósito visa solucionar problemas da Engenharia de Software de forma automática através da utilização de algoritmos de busca.

Posteriormente justificou-se a motivação na opção pela Otimização Interativa, demonstrando seus principais conceitos e limitações. Após essas definições, apresentou-se uma breve análise sobre os trabalhos que exploram os conceitos da Otimização Interativa sob o contexto de SBSE. Em seguida, detalharam-se as peculiaridades e inspirações advindas da técnica de busca utilizada no presente trabalho, o Algoritmo Genético Interativo. Discutiu-se também os conceitos que envolvem a área de Aprendizado de Máquina e as técnicas que serão empregadas no estudo empírico. Em termos de diferença entre o LMS e MLP, constata-se que enquanto no LMS baseia-se nos conceitos da Regressão Linear, no MLP o conhecimento referente ao comportamento entrada/saída do sistema é baseado nos princípios das redes neurais. Por fim, apresentou-se toda motivação e modelagem referente ao Problema do Próximo *Release*.

### 3 TRABALHOS RELACIONADOS

Este capítulo tem como finalidade apresentar os principais trabalhos relacionados à presente pesquisa. A princípio, são discutidos os que estão diretamente relacionados ao Problema do Próximo *Release*. Em seguida expõe-se os que aplicam conceitos da Otimização Interativa na Engenharia de Software Baseada em Busca. Por fim, analisam-se algumas propostas de tratamento da fadiga humana.

#### 3.1 PROBLEMA DO PRÓXIMO RELEASE

Um dos tópicos mais relevantes e utilizados no âmbito de *Search Based Software Engineering* (SBSE) é o Problema do Próximo *Release*. A primeira formulação mono-objetiva foi elaborada por Bagnall et al. (2001). Nessa proposta considera-se a existência de diversos clientes com diferentes níveis de relevância. Cada cliente possui um peso e indica quais requisitos deseja que sejam implementados no próximo *release*. Assim, considera-se um cliente satisfeito quando todos seus requisitos são incluídos no *release*. Dessa forma, objetiva-se selecionar o conjunto de requisitos que maximize a soma dos pesos referentes aos clientes, de tal maneira que a soma de todos os custos dos requisitos selecionados não ultrapassem o orçamento do *release* previamente definido. Com relação à avaliação empírica foram utilizadas métodos exatos, heurísticas de vizinhança e a metaheurística Têmpera Simulada. Em termos de resultados observou-se que para instâncias menores o método exato alcança melhores resultados, porém em instâncias maiores a metaheurística consegue se sobressair.

Diferentemente de Bagnall et al. (2001), Baker et al. (2006) propõe que cada componente (requisito) possua uma certa importância específica atribuída pelos próprios clientes. A importância global de um requisito é calculada através da soma ponderada das importâncias atribuídas por cada cliente. Assim, neste trabalho objetiva-se selecionar um subconjunto de requisitos que maximize a importância global do *release*. Considera-se tal trabalho relevante devido a seus experimentos terem utilizado informações de uma grande empresa de telecomunicações. Com relação a validação, comparou-se os resultados oriundos de algoritmos gulosos, Têmpera Simulada e soluções geradas manualmente por um especialista e, ao fim, concluiu-se que as abordagens automáticas superaram de fato a avaliação humana.

Naturalmente, o NRP já foi explorado através de diversas abordagens. Em (JIANG et al., 2010) propõe-se a utilização de um algoritmo híbrido formado pelo *Ant System* com o

suporte do algoritmo Subida da Colina, cuja principal finalidade consiste em melhorar a solução encontrada. Em (SAGRADO et al., 2010) relata-se a aplicação do *Ant Colony System*, constatando que os resultados alcançados são melhores em qualidade da solução e convergência do que as metaheurísticas Têmpera Simulada e Algoritmo Genético. Complementando tal trabalho, em (FERREIRA; SOUZA, 2012) foi elaborado um estudo comparativo entre os algoritmos *Ant Colony Sistem*, Algoritmo Genético e Têmpera Simulada considerando as interdependências entre os requisitos.

Em (AKKER et al., 2005) aplicam-se técnicas de programação linear inteira considerando aspectos práticos como a lista total de requisitos, as dependências e seus respectivos custos, além do quanto, em termos de recursos, a equipe de desenvolvimento necessita. Um outro trabalho relevante é o (XUAN et al., 2012), pois o mesmo se dedica a resolver grandes instâncias do NRP através de um algoritmo denominado *Backbone Algorithm*. Os resultados alcançados são comparados com uma variação da Têmpera Simulada, conhecida como LMSA.

De maneira geral, todas essas abordagens podem ser consideradas como ferramentas de tomada de decisão, onde sequencialmente o usuário fornece os dados, a ferramenta automatiza o processo e, por fim, retorna um conjunto de requisitos a serem implementados no próximo *release*. Devido à essa automatização, tais abordagens acabam por desconsiderar as preferências humanas durante o processo de busca e conseqüentemente não usufruem de muitos benefícios que o conhecimento humano poderia agregar aos resultados.

### 3.2 OTIMIZAÇÃO INTERATIVA EM ENGENHARIA DE SOFTWARE BASEADA EM BUSCA

Em termos de engenharia de requisitos, pode-se apontar o artigo de Tonella et al. (2010), no qual os autores examinaram a efetividade de um IGA no processo de priorização de requisitos em um estudo de caso real, considerando quesitos como a dependência entre os mesmos. Sua concepção visa minimizar a quantidade de avaliação de “pares de requisitos” obtidas junto ao usuário, fazendo com que a abordagem seja mais escalável e precisa quanto a classificação final dos requisitos. Os resultados demonstraram-se bastante positivos indicando que em sua forma interativa o desempenho é substancialmente melhor, mantendo inclusive a taxa de esforço do usuário numa proporção aceitável. Posteriormente, essa proposta foi estendida em (TONELLA et al., 2013). Dentre as novidades apresentadas, pode-se mencionar a comparação com novos métodos de priorização de requisitos, novos experimentos analisados e melhorias na

descrição do algoritmo proposto.

Já em (BAVOTA et al., 2012) é proposta a integração do desenvolvedor na tarefa de re-modularização de componentes de software. Nesse trabalho optou-se por avaliar automaticamente a qualidade e as dependências entre os módulos, enquanto o usuário fica responsável apenas por avaliar, quando necessário, se dois componentes devem ou não fazer parte do mesmo módulo. Mediante este *feedback*, o IGA penaliza os artefatos que não estão onde deveriam. Apesar da versão automática ser mais efetiva em termos de coesão e redução de módulos, ela não considera o conhecimento dos desenvolvedores na decisão de agrupar ou não determinados componentes.

Ghannem et al. (2013) apresentam uma nova abordagem que sugere sequências de refatoração que possam ser aplicadas em determinado diagrama de classes UML. Para isso, adaptou-se um IGA para explorar exemplos de refatoração previamente existentes em conjunto com o conhecimento do usuário. A abordagem foi implementada como um *plugin* de uma ferramenta de desenvolvimento, cuja entrada é uma base de exemplos de modelos refatorados e um modelo inicial a ser melhorado. A princípio o usuário especifica diversos parâmetros específicos da proposta. Ao fim, é produzida como saída uma sequência ótima de refatorações a serem aplicadas sobre o sistema analisado. Os resultados mostraram que a abordagem é estável com relação a sua exatidão, integridade, tipo e número de refatorações propostas por classe.

Com relação ao contexto de design de software existe uma maciça quantidade de trabalhos aplicando os conceitos de Otimização Interativa, dentre os quais pode-se mencionar (PARMEE et al., 2006), (SIMONS; PARMEE, 2010), (SIMONS et al., 2010), (SIMONS, 2011), (SIMONS; PARMEE, 2012), (SIMONS et al., 2012), (SIMONS; SMITH, 2013). Destaca-se o trabalho (SIMONS et al., 2014), pois o mesmo sintetiza boa parte das ideias e conceitos relatados nos anteriores. Inicialmente a pesquisa utilizava um Algoritmo Genético Interativo, porém posteriormente optou-se pela utilização da metaheurística *Interactive Ant Colony Optimization*, cujo processo de busca é guiado através de um modelo adaptativo que consegue aliar aspectos objetivos e subjetivos. Em termos de interação, o usuário é convidado a prover uma avaliação numérica entre 1 e 100 para uma solução candidata. A representação de uma solução foi elaborada como um diagrama de classe UML. Com relação aos resultados alcançados, os participantes do experimento avaliaram a proposta como convincente, concluindo que a mesma pode ser considerada como uma alternativa interessante na busca para problemas relacionados ao *design* de software.

Sob a perspectiva da área de testes, pode-se mencionar (MARCULESCU et al., 2012), (MARCULESCU et al., 2013) e (MARCULESCU et al., 2014). Todos os três trabalhos fazem parte do mesmo escopo. O objetivo da pesquisa consiste em propor um sistema interativo como suporte para a área de testes de software. Tal sistema faz uma distinção entre os interesses da engenharia de software e do domínio de aplicação, permitindo que os especialistas de domínio interajam com o sistema a fim de selecionar os critérios de qualidade a serem utilizados. É interessante verificar que é considerada apenas a opinião do especialista de domínio, uma vez que cada domínio têm critérios únicos de qualidade e grande variação na modelagem do software.

### 3.3 TRATAMENTO DA FADIGA HUMANA

Com relação à redução da fadiga humana na CEI, Kamalian et al. (2006) propõem, tal qual o presente o trabalho, a utilização de técnicas de inteligência artificial para prever as avaliações humanas com base em resultados anteriores. De maneira geral é proposta a aplicação da CEI no problema de *machine design*. Foram testadas duas abordagens visando diminuir a fadiga humana. A primeira foi um sistema *fuzzy* composto por quatro regras, cujo resultados alcançaram uma redução de 51% do esforço humano. Já a segunda foi a utilização de técnicas de aprendizado de máquina, na qual atingiu-se uma redução de 31%.

Já Hsu e Huang (2005) acreditam que uma das principais causas desse problema se deve ao fato de que existem ocasiões cujo resultado preferido pelo cliente pode não existir no espaço de busca. Logo, se não é possível garantir que a solução idealizada pelo usuário existe no espaço de busca, a tendência é que seja mais difícil de se alcançar a mesma e que haja um esforço ineficiente. Diante dessa concepção, é proposto um modelo de CEI que garanta ao usuário um espaço de busca mais completo. Um estudo de caso envolvendo o design de garrafas de água mineral foi analisado visando verificar a capacidade do modelo. De modo geral, os resultados obtidos confirmam a conjectura de que um espaço de busca correto contribui na redução da fadiga.

Uma outra alternativa explorada por Wang et al. (2006) é a criação de uma escala absoluta para melhorar a predição de avaliações humanas e acelerar a convergência do algoritmo evolucionário a ser utilizado, reduzindo assim, a quantidade de intervenções humanas. O algoritmo proposto foi aplicado em um sistema de recuperação de imagens e apresentou bons resultados em termos de performance, além de aliviar consideravelmente a fadiga. Os resultados demonstraram que a velocidade de convergência de um sistema CEI utilizando a concepção

apresentada é mais rápida do que se fosse utilizado convencionalmente.

### 3.4 CONCLUSÕES DO CAPÍTULO

Este capítulo expôs os trabalhos encontrados na literatura relacionados à presente pesquisa. A apresentação dos mesmos seguiu a seguinte ordem: primeiramente discutiu-se alguns dos principais trabalhos correlacionados ao Problema do Próximo *Release*. Pode-se verificar que o processo de seleção de requisitos para o próximo *release* já foi modelado de diversas formas e, conseqüentemente, varias técnicas foram propostas como forma de resolução.

Em seguida analisou-se os trabalhos que envolvem a aplicação de conceitos de Otimização Interativa em Engenharia de Software Baseada em Busca. Averiguou-se que muitos tópicos já foram exploradas como, engenharia de requisitos, re-modularização de componentes, refatoração, testes e *design* de software. Dentre os algoritmos mais utilizados destaca-se o IGA e *Interactive Ant-Colony Optimization*.

Por fim, dedicou-se um espaço visando apresentar três abordagens distintas para tratamento da fadiga humana em contextos que envolvem a CEI. Todos são enfáticos ao retratar a fadiga como um dos principais problemas a serem discutidos pela comunidade.

É válido mencionar que, mediante uma revisão da literatura, constatou-se que até o momento não existe nenhum trabalho que explore o NRP através da aplicação de um IGA.

## 4 ARQUITETURA E FORMULAÇÃO INTERATIVA PARA O NRP

Neste capítulo definem-se todos os aspectos e relações que compõe a arquitetura elaborada. Em seguida, apresenta-se uma modelagem interativa para o Problema do Próximo *Release* sob a perspectiva da arquitetura proposta. Ao fim, exemplifica-se através de um cenário de aplicação da abordagem.

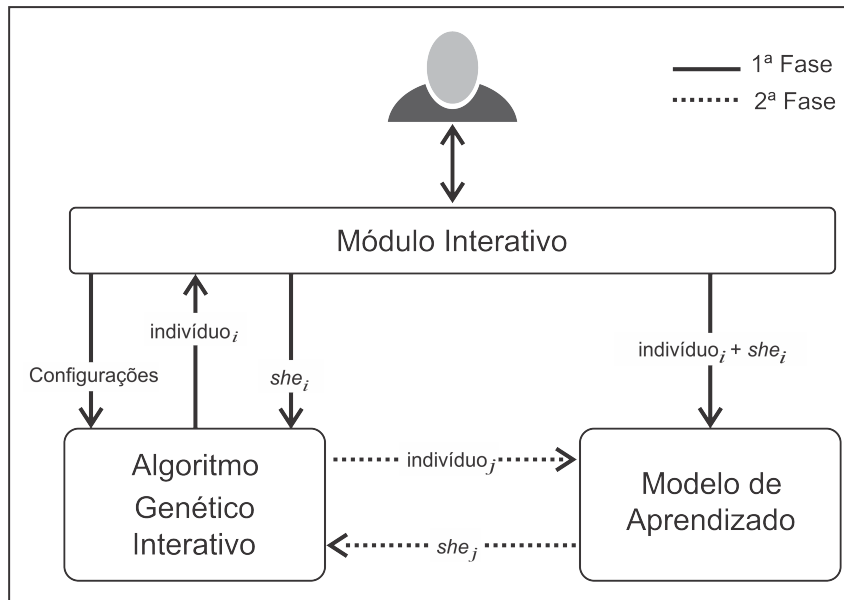
### 4.1 VISÃO GERAL DA ARQUITETURA

Esse trabalho propõe uma arquitetura capaz de incorporar as preferências do usuário no processo de busca através de um Algoritmo Genético Interativo. Para contornar o problema da fadiga humana, utiliza-se um modelo de aprendizado de máquina responsável por aprender o perfil do usuário a fim de substituí-lo na avaliação subjetiva das soluções. Conforme verifica-se na Figura 15, a arquitetura é composta por três componentes que correspondem respectivamente ao:

- a) **Algoritmo Genético Interativo:** responsável pelo processo de otimização. Nele são geradas as soluções que, a princípio, serão avaliadas por intermédio do componente *Módulo Interativo* e, posteriormente, pelo *Modelo de Aprendizado*. A aptidão dos indivíduos é calculada considerando o aspecto matemático específico do problema em conjunto com a avaliação subjetiva humana que é encapsulada numa variável especial denominada *she*.
- b) **Módulo Interativo:** é a interface de interação com o usuário. A cada interação, um indivíduo oriundo do IGA é apresentado ao usuário, que por sua vez terá a responsabilidade de fornecer sua avaliação subjetiva humana (*she*). Para o IGA será enviado apenas o *she*, enquanto para o *Modelo de Aprendizado* será compartilhado o *she* e sua respectiva solução. Além disso, o usuário também deve definir a priori algumas configurações da arquitetura.
- c) **Modelo de Aprendizado:** responsável por aprender o perfil de avaliação do usuário através da utilização de uma técnica de aprendizado de máquina previamente definido. Neste componente ocorre o processo de treinamento de acordo com as amostras (indivíduos e *she*) advindas do *Módulo Interativo*. Após a conclusão do treinamento, o modelo estará apto a substituir o ser humano na avaliação das soluções.

Apresenta-se na Figura 15 uma visão geral da arquitetura proposta, incluindo os componentes mencionados anteriormente:

Figura 15 – Visão Geral da Arquitetura



Fonte: Elaborada pelo autor.

A relação entre estes componentes é dividida em duas fases, conforme distinguido a seguir:

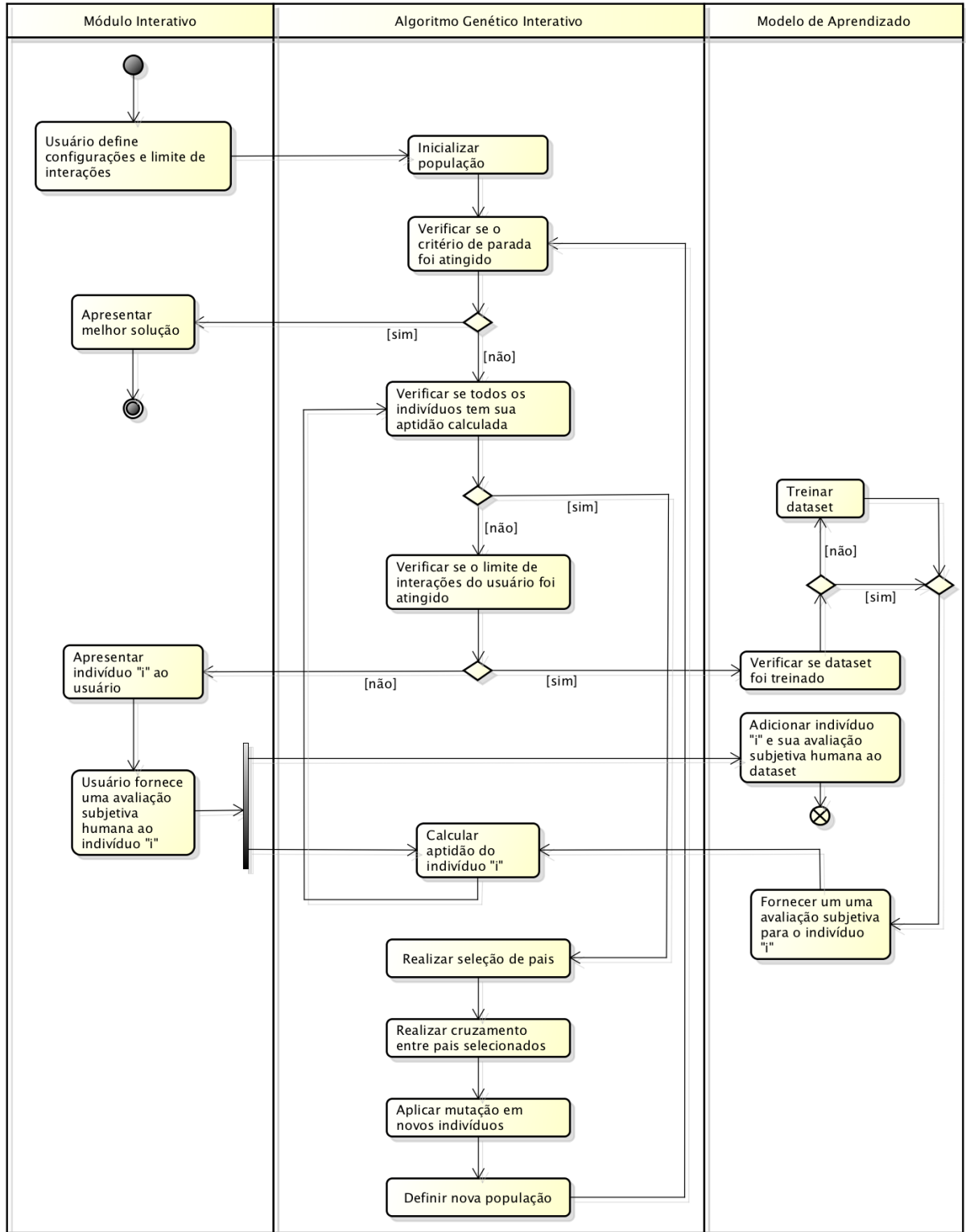
- a) **Primeira fase (linhas sólidas):** todas as soluções são avaliadas pelo usuário. O IGA é guiado de acordo com suas preferências enquanto cada indivíduo e sua respectiva avaliação subjetiva humana ( $she$ ) são enviados ao *Modelo de Aprendizado*.
- b) **Segunda fase (linha pontilhada):** neste momento inicia-se o treinamento do modelo a partir do conjunto de amostras colhido na fase anterior. Após a conclusão desse processo, o *Modelo de Aprendizado* terá a responsabilidade de simular as predileções do usuário e emitir uma avaliação subjetiva humana ( $she$ ) para cada indivíduo.

Na Figura 16 é possível visualizar o fluxo da arquitetura através de um Diagrama de Atividades que representa o funcionamento do *Algoritmo Genético Interativo* aliado ao *Módulo Interativo* e *Modelo de Aprendizado*. Conforme supracitado, o fluxo seguirá primordialmente os preceitos da versão canônica de um Algoritmo Genético, mas é preponderante que o ser humano defina a priori algumas configurações arquiteturais. Dessa forma, o usuário deve especificar o



quão influente será o aspecto subjetivo no processo de busca e quanto o mesmo está disposto a interagir.

Figura 16 – Diagrama de Atividades da Arquitetura



Fonte: Elaborada pelo autor.

Primeiramente, o usuário deve configurar o impacto do aspecto subjetivo no processo de busca e a quantidade de perguntas que o mesmo está disposto a responder. Após isso, inicializa-se a população inicial e verifica-se se o critério de parada do algoritmo foi atingido. Caso o critério não tenha sido atingido, calcula-se a aptidão de todos os indivíduos da população. Essa etapa é baseada num limite de interações com o usuário e pode ocorrer de duas formas:

- a) Caso o limite de interações *não tenha* sido atingido, apresenta-se o indivíduo ao usuário o qual tem a responsabilidade de atribuir uma avaliação subjetiva. Por sua vez, tal avaliação será incorporada no cálculo da aptidão do indivíduo, além de, em conjunto com seu respectivo indivíduo, serem adicionados ao *dataset* de treinamento.
- b) Caso o limite de interações *tenha* sido atingido, verifica-se se o *dataset* foi devidamente treinado. Se o *dataset* ainda não foi treinado, efetua-se seu treinamento. Caso contrário, o *Modelo de Aprendizado* deve fornecer uma avaliação subjetiva para cada indivíduo a ser avaliado.

Ao se calcular a aptidão de todos os indivíduos da população, aplicam-se os operadores genéticos e, conseqüentemente, prossegue-se o fluxo do algoritmo até que o critério de parada seja atingido, ou seja, caso ainda não tenha sido atingido, o algoritmo continuará sendo executado considerando apenas as avaliações subjetivas provenientes do *Modelo de Aprendizado*. A seguir será detalhada a formulação necessária para adequação do Problema do Próximo *Release* ao requisitos da arquitetura proposta.

## 4.2 UMA FORMULAÇÃO INTERATIVA PARA O NRP

A versão canônica do NRP não considera de maneira incisiva a opinião humana no processo de avaliação das soluções. Dito isso, o presente trabalho propõe uma formulação interativa para o NRP que possibilite ao mesmo ser resolvido pela arquitetura elaborada. Essa formulação, chamada Interactive Next Release Problem (iNRP), é proporcionada através da Modelagem Matemática e Modelagem Interativa.

### 4.2.1 Modelagem Matemática

Considere  $R = \{r_1, r_2, r_3, \dots, r_N\}$  o conjunto de todos requisitos independentes e possíveis de serem selecionados para o próximo *release*, onde  $N$  representa o número de requisi-

tos. Cada requisito  $r_i$  possui um valor de importância total  $v_i$  e um custo de desenvolvimento  $e_i$ . Considere também  $C = \{c_1, c_2, c_3, \dots, c_M\}$  o conjunto de clientes atendidos preço pelo sistema, onde cada cliente  $c_j$  terá um determinado peso  $w_j \in Z^+$  que mensura sua respectiva importância. Logo, a modelagem matemática para o iNRP proposta nesse trabalho é a seguinte:

$$\begin{aligned} &\text{maximizar: } \alpha \times \text{score}(X) + \beta \times \text{she}(X) \\ &\text{sujeito a: } \text{cost}(X) \leq \text{budget} \end{aligned} \quad (4.1)$$

$$\text{onde, } \text{score}(X) = \sum_{i=1}^N v_i \times x_i \quad (4.2)$$

$$v_i = \sum_{j=1}^M w_j \times s_{ji} \quad (4.3)$$

$$\text{cost}(X) = \sum_{i=1}^N e_i \times x_i \quad (4.4)$$

onde o *budget* refere-se ao orçamento disponível para o *release*. Utiliza-se uma representação de solução binária, onde um *release* é representado pelas variáveis de decisão  $X = \{x_1, x_2, x_3, \dots, x_N\}$ , onde  $x_i = 1$  indica que o requisito  $r_i$  foi incluso no próximo *release* e  $x_i = 0$  o contrário. A função *score* (Equação 4.2) do *release* é calculada pelo somatório do produto entre a importância total ( $v_i$ ) do requisito  $r_i$  e a variável de decisão  $x_i$ . A importância total ( $v_i$ ) de selecionar um requisito  $r_i$  é dada pelo somatório dos produtos dos pesos ( $w_j$ ) de cada cliente pela importância específica ( $s_{ji}$ ) que cada um fornece ao requisito  $r_i$  (Equação 4.3). De forma análoga ao *score*, a função *cost* (Equação 4.4) representa o custo total de desenvolvimento do *release* e é calculada pelo somatório do produto entre o custo ( $e_i$ ) do requisito  $r_i$  e a variável de decisão  $x_i$ .

Na presente aplicação do IGA para a versão interativa do NRP, optou-se por representar cada indivíduo como um *release*. Dessa forma, o usuário fornecerá uma “nota” para cada indivíduo avaliado ao longo da evolução do IGA. Essa “nota” será o valor referente a variável *she* e, conseqüentemente, representará suas preferências em relação à seleção de requisitos. Esse valor será dado de acordo com um intervalo numérico previamente definido. Quando o *release* satisfaz totalmente o usuário, a nota é máxima. Similarmente, a avaliação é mínima quando o *release* é completamente diferente do que o mesmo deseja. E, conseqüentemente, proporcionalmente para as outras possibilidades.

A abordagem utilizada nesse trabalho pode ser considerada como uma generalização do trabalho de Baker *et al.* (2006). Quando os pesos na Equação 4.1 são configurados para  $\alpha = 1$  e  $\beta = 0$ , o NRP clássico é atingido, ou seja, a seleção de requisitos será feita considerando somente a função de *score*. Quando  $\alpha = 0$  e  $\beta = 1$ , apenas as preferências humanas serão consideradas no processo de busca. Já a configuração  $\alpha = 1$  e  $\beta = 1$  considera as preferências humanas e, ao mesmo tempo, busca selecionar os requisitos com maior *score*. Essa liberdade de configuração dos pesos permite ao usuário estabelecer antes da execução do algoritmo qual opção melhor se enquadra ao cenário confrontado. Por exemplo, caso necessite-se de uma solução que priorize duas vezes mais a opinião humana do que o *score* basta se optar pela configuração  $\alpha = 1$  e  $\beta = 2$ . De maneira geral, tal cálculo é sintetizado através do seguinte algoritmo:

---

**Algoritmo 2:** Cálculo da Aptidão do Indivíduo no IGA

---

**Entrada:** Indivíduo, *she*,  $\alpha$ ,  $\beta$

**Saída:** Aptidão do indivíduo

**início**

    Calcula o *score* do indivíduo;

    Normaliza o *score* do indivíduo para a mesma escala da sua *she*;

    Aptidão do indivíduo  $\leftarrow (\alpha \times \textit{score}) + (\beta \times \textit{she})$ ;

**fim**

---

No presente trabalho optou-se por normalizar o valor de *score* para o mesmo intervalo de *she* a fim de se evitar um possível desbalanceamento entre tais funções. A partir dessa normalização, a única forma de priorizar uma das funções em relação à outra é a partir dos valores concedidos ao  $\alpha$  e  $\beta$  na função objetivo. A formulação matemática referente a normalização do  $\textit{score}(X)$  é descrita a seguir:

$$n\textit{Score}(X) = \left( \frac{\textit{score}(X)}{\textit{score}_{max}} \right) * \textit{she}_{max} \quad (4.5)$$

onde,  $X$  é uma solução,  $\textit{score}_{max}$  é o maior valor em termos de *score* que uma solução pode possuir e  $\textit{she}_{max}$  é o maior valor que a avaliação humana pode assumir.

## 4.2.2 Modelagem Interativa

Ao se optar por utilizar os conceitos de Otimização Interativa é natural que seja necessário consolidar todo o processo que envolve a interação humana com o sistema. Essa etapa é muito importante para a arquitetura, haja vista que está estrategicamente conectada aos requisitos específicos do problema e as necessidades do usuário. Conforme proposto na seção 2.3, algumas perguntas são relevantes de serem discutidas e definidas para que ocorra uma melhor adaptação ao problema, são elas:

a) **Em qual momento a preferência será capturada?**

Na presente abordagem optou-se por incluir a avaliação subjetiva humana durante a evolução do algoritmo, proporcionando assim, que as soluções futuras sejam devidamente influenciadas pelas preferências humanas. Logo, necessita-se estipular previamente uma determinada quantidade de indivíduos a ser avaliada pelo usuário, uma vez que a construção do conjunto de treinamento depende dos mesmos e, conseqüentemente, pode afetar os resultados do modelo de aprendizado.

b) **Qual informação será concedida pelo usuário?**

Conforme definido na arquitetura, a informação proveniente do usuário para cada solução é representada através da variável *she*, cujo principal objetivo consiste em representar seus critérios subjetivos acerca de um determinado conjunto de requisitos. Optou-se neste estudo pela *she* ser refletida através de uma nota que obedeça um certo intervalo numérico.

c) **De que forma a opinião será incorporada e influenciará no processo de busca?**

A opinião do usuário será englobada na função de aptidão como um objetivo específico, logo, todas as soluções serão influenciadas pela percepção humana. Assim, o processo de busca estará apto a guiar a construção dos indivíduos de acordo com as preferências do usuário.

## 4.3 EXEMPLO DE APLICAÇÃO

Todo o comportamento que envolve a abordagem proposta será exemplificado através de um cenário de aplicação, conforme pode-se averiguar a seguir:

Considere uma empresa cujo sistema *Enterprise Resource Planning* (ERP) é mantido

pelo seu setor de TI particular. A equipe técnica emprega no seu processo de desenvolvimento a metodologia incremental e necessita saber quais requisitos deverão ser implementados no próximo *release* do sistema, cujo *budget* foi definido em R\$ 7.200,00. O cenário enfrentado é composto por seis requisitos e quatro clientes, que representam diferentes departamentos da organização. Naturalmente, tais clientes possuem prioridades e níveis de relevância distintos, logo, optou-se por quantificar a importância de cada um através do estabelecimento de um peso. Cada cliente também terá a incumbência de mensurar uma determinada importância para os requisitos que podem ser implementados. O Quadro 1 sintetiza o cenário em discussão.

Quadro 1 – Visão Geral do Cenário

Cliente	Peso	$r_1$ CRUD Funcionários (Custo: 1.000)	$r_2$ CRUD Fornecedores (Custo: 4.000)	$r_3$ Autenticação Via Login (Custo: 2.000)	$r_4$ Pesquisa de Produtos (Custo: 3.000)	$r_5$ Visualização de Estoque (Custo: 2.000)	$r_6$ Relatórios Gerenciais (Custo: 1.000)
Marketing	0.4	2	3	1	2	3	5
RH	0.2	1	2	2	5	3	1
Financeiro	0.1	5	1	4	2	1	4
Vendas	0.3	2	2	2	3	1	4

Fonte: Elaborado pelo autor.

Destaca-se ainda, a presença de um engenheiro de requisitos o qual terá como responsabilidade representar e estabelecer o melhor *trade-off* entre as preferências relacionadas aos quatro clientes do software. O mesmo estará apto para incorporar tais aspectos e avaliar as soluções oriundas do IGA, dado seu conhecimento de domínio e visão estratégica. Antes de executar o algoritmo, o engenheiro configurou um limite máximo de 100 perguntas e os pesos da função objetivo em  $\alpha = 1$  e  $\beta = 1$ , ou seja, no processo de busca os dois objetivos serão equivalentes.

Assim, inicia-se a execução do algoritmo onde posteriormente é calculada a importância total de se selecionar determinado requisito. Este cálculo consiste no somatório do produto entre os pesos dos clientes e valor de importância específica que estes atribuíram ao requisito  $r_i$ . Por exemplo, a importância total de se incluir o “CRUD Funcionários” é dada por  $(0,4 * 3) + (0,2 * 1) + (0,1 * 5) + (0,3 * 2) = 2,1$ . Analogamente, o valor referente ao restante dos requisitos será respectivamente  $r_2 = 2,3$ ,  $r_3 = 1,8$ ,  $r_4 = 2,9$ ,  $r_5 = 2,2$ , e  $r_6 = 3,8$ . Logo, verifica-se por exemplo que “Relatórios Gerenciais” seria o requisito com maior importância total, enquanto a “Autenticação via Login” a menor.

No entanto, os valores a serem otimizados pela função objetivo proposta nesse trabalho são o *score* e *she*. Para se obter o *score*, basta calcular o somatório da importância total

dos requisitos inclusos no indivíduo a ser avaliado, ou seja, em uma possível solução [101101], seu valor de *score* seria  $(2,1 * 1) + (2,3 * 0) + (1,8 * 1) + (2,9 * 1) + (2,2 * 0) + (3,8 * 1) = 10,6$ . Verificando a conjuntura do *release*, o engenheiro de requisitos julgou que a solução está boa devido a presença do “Relatórios Gerenciais” e forneceu uma *she* valorada em 80, cujo intervalo numérico ficou estabelecido em no mínimo 0 e no máximo 100. O mesmo tem consciência que se tivesse “Visualização de Estoque” sua nota seria ainda superior, afinal, ambos requisitos poderiam contribuir para aspectos relacionados a tomada de decisão da organização. Normalizando o valor *score* e somando com sua respectiva *she*, o valor de sua aptidão seria  $(1 * 70,1) + (1 * 80) = 150,1$ .

Lembrando que tal solução seria válida pois consegue atender a restrição do problema, haja vista que a soma dos custos individuais dos requisitos incluídos  $((1.000 * 1) + (4.000 * 0) + (2.000 * 1) + (4.000 * 1) + (3.000 * 0) + (2.000 * 0) + (1.000 * 1) = \text{R\$ } 7.000)$  obedece o orçamento disponível para o *release*.

Esse processo de interação com o usuário é repetido até completar as 100 interações previamente estipuladas. Isso quer dizer que os primeiros 100 indivíduos da população obterão um *she* fornecido pelo próprio ser humano. Conforme explanado anteriormente, essas soluções e suas respectivas avaliações serão incluídas no conjunto de treinamento para posteriormente serem submetidos ao processo de treinamento de acordo com a técnica especificada no *Modelo de Aprendizado*.

Dar-se início a segunda fase onde efetiva-se o processo de treinamento e todo o restante dos indivíduos são avaliados de acordo com o *Modelo de Aprendizado*. Assim, possibilita-se que a *she* para cada possível solução continue condizente com o perfil do usuário aprendido na primeira fase. Este ciclo será encerrado quando se atingir um total de 100 gerações do IGA. Ao fim, devido a configuração de pesos, retorna-se uma solução final equilibrada tanto em termos de *score*, quanto *she*. Portanto, será possível obter um conjunto de requisitos que se adeque às preferências humanas e, ainda assim, continue bom do ponto de vista objetivo do problema.

#### 4.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo apresentou-se toda configuração necessária para aplicação da arquitetura proposta. Justifica-se tal abordagem dado o benefício que incluir o ser humano no processo de busca pode trazer aos resultados.

Primeiramente definiu-se os três principais componentes da arquitetura. O Algoritmo Genético Interativo é responsável pelo processo de otimização. O mesmo segue as características de sua versão canônica, com a diferença de que a opinião humana é incorporada na avaliação das soluções aliada ao aspecto matemático específico do problema explorado. O outro componente é o *Módulo Interativo*, que além de ser a interface de interação com usuário, tem a responsabilidade de compartilhar suas preferências com o *Modelo de Aprendizado*. Por fim, o próprio *Modelo de Aprendizado*, no qual objetiva-se aprender o perfil de avaliação do usuário através de uma técnica de aprendizado e posteriormente subsidiar a automatização das decisões.

Cada componente tem sua respectiva responsabilidade e a necessidade de funcionar complementarmente em duas fases distintas. De maneira sucinta, na primeira etapa apenas o ser humano avalia as soluções e proporciona ao modelo amostras de treinamento. Na segunda fase treina-se modelo de acordo com as amostras coletadas, e assim, possibilita-se ao mesmo ser capaz de fornecer uma avaliação subjetiva condizente com o modelo aprendido.

Após essas definições, apresentou-se a concepção de uma formulação interativa para o Problema do Próximo *Release*, denominada como iNRP. Essa adaptação é viabilizada através de duas modelagens distintas responsáveis por adequar o problema à arquitetura e, consequentemente, incorporar a presença humana no processo de avaliação das soluções. Primeiramente foram especificados os detalhes que envolvem a modelagem matemática da aplicação do IGA para o iNRP. Em seguida, foram respondidos os questionamentos que definem toda a estratégia de interação a ser utilizada.

Por fim, exemplificou-se um cenário de aplicação no qual elucida as principais peculiaridades da proposta, possibilitando, assim, uma total compreensão sobre a abordagem. Portanto, ao definir uma arquitetura para resolução do NRP atinge-se o objetivo principal deste trabalho e, ao se elaborar uma formulação interativa para o NRP alcança-se o objetivo específico “a” devidamente explicitado no Capítulo 1.

No próximo capítulo, será mostrado o estudo empírico conduzido para validação e avaliação da abordagem em prol de uma análise mais detalhada sobre os resultados gerados.



## 5 ESTUDO EMPÍRICO

O propósito deste estudo empírico consiste em responder as perguntas de pesquisa elencadas no objetivo específico que propõe o teste e análise comportamental da abordagem proposta. Os tópicos a seguir apresentam as instâncias, configurações, métricas elaboradas e os respectivos resultados e análises. Visando facilitar tais análises, os experimentos realizados objetivam responder quatro questões de pesquisa que são devidamente descritas posteriormente.

### 5.1 INSTÂNCIAS

O conjunto de instâncias utilizadas é constituído por dados artificiais, cuja geração foi elaborada de modo aleatório. O número de requisitos varia entre 50, 100, 150 e 200. Não existe interdependências entre os requisitos e a importância específica de cada requisito assume um valor inteiro entre 1 e 5. A quantidade de clientes também é gerada de maneira aleatória. O nome das instâncias segue o formato I\_R, onde R representa o número de requisitos. Exemplificando, se uma instância tem 50 requisitos, ela seria nomeada como I\_50. Para todas as instâncias o orçamento considerado nos experimentos foi de 60% do custo total do *release*.

### 5.2 CONFIGURAÇÕES

Com a finalidade de representar o papel de um usuário, um simulador foi desenvolvido. O principal propósito desse simulador não é simular fielmente um ser humano, mas demonstrar a influência de um perfil de avaliação sobre o processo de busca. Dessa forma, o simulador define um “indivíduo-alvo” o qual representa o que o ser humano “consideraria” como *release* ideal. Os requisitos serão incluídos aleatoriamente no indivíduo-alvo com base em uma certa porcentagem, haja vista que existe uma restrição de custo a ser obedecida e não é possível simplesmente incluir todos os requisitos. Nos testes realizados optou-se por uma taxa de 50% de inclusão, ou seja, em um *release* constituído de 6 requisitos, apenas 3 seriam incluídos aleatoriamente.

Ao longo do IGA, as avaliações subjetivas para cada indivíduo são fornecidas de acordo com o número de requisitos similares ao indivíduo-alvo, ou seja, se seus requisitos são totalmente diferentes do indivíduo-alvo a nota é mínima, já quando a solução possui os mesmos requisitos do indivíduo-alvo a nota é máxima, e assim proporcionalmente para as outras possibilidades. Em particular, nesse estudo empírico definiu-se como *she* mínimo igual a 0 e

máximo 100. Considere, como exemplo, um conjunto de 6 requisitos com um indivíduo-alvo [100011]. Uma possível solução [110110] apresenta 3 requisitos iguais ao indivíduo-alvo ( $r_1$ ,  $r_3$  e  $r_5$ ). Logo, se todos os 6 requisitos da solução candidata fossem iguais aos do indivíduo-alvo a nota seria 100. Então, para três requisitos iguais, a avaliação subjetiva humana ( $she$ ) para essa solução em específico será 50. A formulação matemática para tal cálculo é dada por:

$$she(X, P) = \left( \frac{\sum_{i=1}^N x_i p_i}{N} \right) \times she_{max} \quad (5.1)$$

onde,  $X$  é uma solução,  $N$  é o total de requisitos presentes na solução,  $x_i$  indica se o requisito  $r_i$  está ou não presente na solução,  $p_i$  indica se o requisito  $r_i$  pertence ou não ao indivíduo-alvo  $P$  e  $she_{max}$  é o valor máximo que a avaliação subjetiva humana pode assumir. Para o presente estudo empírico, optou-se pelo mínimo  $she(X, P)$  ser 0 e o máximo 100.

Todas as configurações do IGA foram obtidas empiricamente a partir de testes preliminares. São elas:

- a) Número de indivíduos é o dobro da quantidade de requisitos;
- b) 100 gerações;
- c) Taxa de cruzamento 90%;
- d) Taxa de mutação  $1/N$ , sendo  $N$  o número de requisitos;
- e) 20% de elitismo.

Com relação aos detalhes referentes ao aprendizado de máquina, optou-se por duas técnicas com características distintas. Enquanto o *Least Median Square* (LMS) caracteriza-se por ser um técnica de aprendizado baseado no modelo de Regressão Linear, a *Multilayer Perceptron* (MLP) é reconhecidamente uma das Redes Neurais mais utilizadas. Dessa forma, contribui-se para uma visão mais consolidada e abrangente da arquitetura proposta. Em termos de parametrização, utilizou-se os valores padrão sugeridos pela ferramenta *WEKA* (HALL et al., 2009).

Para cada configuração de pesos, quantidade de interações com o usuário, instância e técnica de aprendizado de máquina, o IGA foi executado 30 vezes coletando-se a média e o desvio padrão dos valores obtidos. Todas as instâncias e resultados do estudo empírico, inclusive os que tiveram de ser omitidos por restrições de espaço estão disponíveis na página <<http://goes.uece.br/allyssonaraujo/architecture4inrp/>> e nos Apêndices A e B desta dissertação.

### 5.3 MÉTRICAS

Para se promover uma análise mais significativa necessitou-se elaborar algumas métricas que esclareçam devidamente os resultados alcançados e, conseqüentemente, contribuam no embasamento relacionado às respostas das questões de pesquisa. Considera-se que tais métricas sejam genéricas suficientes para serem utilizadas em outros trabalhos que explorem os conceitos de Otimização Interativa em SBSE.

#### 5.3.1 Grau de Similaridade

O *Grau de Similaridade (GS)* representa o quão similar, em termos percentuais, é uma solução candidata em relação ao indivíduo-alvo. Considere um conjunto de 6 requisitos com um indivíduo-alvo [100011]. Uma possível solução [110110], por exemplo, apresenta três requisitos iguais ao indivíduo-alvo ( $r_1$ ,  $r_3$  e  $r_5$ ), conseqüentemente, tal solução teria um *GS* de  $3/6 = 0,5$ , que multiplicando por 100 obtêm-se um valor de 50% de *GS*. Tal resultado é obtido através da seguinte equação:

$$GS(X, P) = \left( \frac{\sum_{i=1}^N x_i p_i}{N} \right) \times 100 \quad (5.2)$$

onde,  $X$  é uma solução para a qual se deseja calcular *GS* em relação a um indivíduo-alvo  $P$ ,  $N$  é o total de requisitos presentes na solução,  $x_i$  indica se o requisito  $r_i$  está ou não presente na solução e  $p_i$  indica se o requisito  $r_i$  pertence ou não ao indivíduo-alvo.

#### 5.3.2 Fator de Similaridade

O *Fator de Similaridade (FS)* indica a diferença percentual em *GS* quando se compara uma solução com e sem influência humana. Por exemplo, considere duas soluções. A primeira solução com intervenção humana apresenta um *GS*( $Y$ ) de 85,33%. Já a segunda, sem influência humana, possui um *GS*( $X$ ) de 54,27%. Mesmo sem considerar a avaliação subjetiva humana (*she*) no processo de busca é normal por volta de 50%, haja vista que podem coincidir de alguns dos seus requisitos serem similares aos do indivíduo-alvo. Dessa forma, o ganho em *FS* entre

essas duas soluções é de 57,2%. Esse valor é dado por:

$$FS(Y,X) = \left( \frac{GS(Y)}{GS(X)} - 1 \right) \times 100 \quad (5.3)$$

onde,  $Y$  é a solução gerada interativamente, ou seja, o processo de busca foi influenciado por preferências humanas e  $X$  é a solução gerada sem considerar avaliações subjetivas humanas.

### 5.3.3 Preço da Preferência

O *Preço da Preferência (PP)* retrata o quanto é perdido percentualmente, em termos objetivos, para se incorporar as preferências do usuário. Considere duas soluções. A primeira sem influência humana cujo valor  $score(X)$  foi de 115,87. Já a segunda, com intervenção humana, apresenta um  $score(Y)$  de 99,89. É natural que haja esta perda visto que muitos dos requisitos preferíveis pelo ser humano podem não ser bons matematicamente, apenas subjetivamente. Logo, a perda em *PP* entre essas duas soluções é de 16,0%. Tal valor é obtido por:

$$PP(X,Y) = \left( 1 - \frac{score(X)}{score(Y)} \right) \times 100 \quad (5.4)$$

onde,  $X$  é a solução totalmente automática, ou seja, gerada sem interferência humana e  $Y$  é a solução gerada considerando-se além do aspecto objetivo, as avaliações subjetivas humanas.

## 5.4 RESULTADOS E ANÁLISES

Os resultados referentes ao estudo empírico são apresentados nesta seção. Visando facilitar a compreensão, cada tópico possui uma determinada questão de pesquisa (*QP*) a ser analisada.

### 5.4.1 Relação entre quantidade de perguntas e preferências atendidas

Esta análise consiste em verificar a relação entre a quantidade de perguntas respondidas pelo usuário e a métrica *Grau de Similaridade (GS)*, ou seja, a quantidade preferências atendidas. Dessa forma, a questão de pesquisa referente a este tópico na qual objetiva-se responder é:

**QP<sub>1</sub>: Qual a relação entre a quantidade de perguntas submetidas ao ser humano e o total de preferências atendidas na solução final?**

A Tabela 1 apresenta os valores médios e desvios padrões de *GS* para diferentes quantidades de perguntas, considerando todas as instâncias artificiais e os dois algoritmos de aprendizado de máquina. Pode-se verificar que, conforme o aumento na quantidade de perguntas, o valor referente ao *GS* tende a crescer para todas instâncias executadas e para ambas técnicas de aprendizado de máquina. É válido ressaltar que, dependendo do tamanho da instância, a quantidade de perguntas tende a ser maior, ou seja, a quantidade de interações humanas necessárias para estabilizar é proporcional ao tamanho da instância utilizada.

Tabela 1 – Valores referentes ao *Grau de Similaridade* para diferentes quantidades de perguntas utilizando LMS e MLP

Perguntas	I_50		I_100		I_150		I_200	
	LMS	MLP	LMS	MLP	LMS	MLP	LMS	MLP
10	51,93 ± 3,33	57,53 ± 4,02	59,27 ± 2,38	59,63 ± 2,59	56,4 ± 1,95	58,47 ± 2,55	54,13 ± 2,12	54,5 ± 2,04
20	54,27 ± 5,77	64,93 ± 4,28	59,07 ± 1,63	63,37 ± 3,15	57,27 ± 1,86	60,78 ± 2,24	54,27 ± 2,45	56,17 ± 2,7
30	60,2 ± 5,85	66,87 ± 3,96	60,63 ± 2,87	65,9 ± 3,47	59,93 ± 3,1	62,87 ± 3,05	55,65 ± 2,66	58,53 ± 2,07
40	68,87 ± 6,98	74,27 ± 4,06	63,67 ± 4,25	67,53 ± 2,94	62,04 ± 3,27	64,2 ± 2,55	56,85 ± 2,94	60,05 ± 2,12
50	84,93 ± 1,24	76 ± 3,06	65,6 ± 4,62	70,5 ± 1,96	63,87 ± 3,11	66,42 ± 2,44	58,57 ± 3,02	61,33 ± 2,21
60	85,33 ± 1,74	77,2 ± 2,86	71,43 ± 4,4	71,93 ± 2,53	66,02 ± 3,63	68,2 ± 2,71	60,2 ± 2,43	62,07 ± 2,23
70	85,27 ± 0,96	77,53 ± 2,11	73,07 ± 4,81	74 ± 2,84	67,18 ± 3,76	69,29 ± 2,22	63,18 ± 4,08	63,78 ± 2,09
80	85,47 ± 0,88	78,2 ± 3,16	76,77 ± 3,84	74,77 ± 2,32	69,96 ± 3,89	70,73 ± 2,81	64,73 ± 2,97	63,82 ± 3,23
90	85,67 ± 0,75	79,27 ± 2,71	80,27 ± 3,38	76,1 ± 2,61	73,07 ± 3,3	72,02 ± 2,58	65,25 ± 3,2	64,48 ± 2,27
100	85,27 ± 0,96	79 ± 2,05	82,4 ± 0,71	77,53 ± 2,17	75,49 ± 3,19	72,84 ± 2,78	69,23 ± 3,48	66,15 ± 2,67
200	85,47 ± 0,88	80,8 ± 2,45	82,3 ± 0,53	80,57 ± 1,36	86 ± 0,77	78,38 ± 1,84	74,75 ± 7,48	72,07 ± 1,91
300	85,47 ± 0,88	81,6 ± 2,5	82,33 ± 0,6	80,43 ± 1,5	86,27 ± 0,87	78,82 ± 1,49	85,22 ± 0,77	73,12 ± 1,69
400	85,33 ± 0,94	82,6 ± 2,48	82,57 ± 0,62	80,77 ± 1,05	86,09 ± 0,75	78,49 ± 1,88	85,25 ± 0,78	72,85 ± 1,86
500	85,67 ± 0,75	83,73 ± 2,11	82,3 ± 0,64	81,37 ± 1,02	86,38 ± 0,78	76,98 ± 2,22	85,33 ± 0,73	72 ± 1,59
600	85,53 ± 0,85	84,27 ± 1,34	82,37 ± 0,66	81,07 ± 1,03	86,22 ± 0,8	77,29 ± 2,22	85,3 ± 0,67	73,13 ± 1,77
700	85,67 ± 0,75	83,73 ± 1,98	82,2 ± 0,54	81,37 ± 0,8	86,33 ± 0,82	77,31 ± 2,36	85,23 ± 0,8	72,3 ± 2,18
800	85,47 ± 0,88	84,33 ± 1,72	82,37 ± 0,48	81,67 ± 0,75	86,24 ± 0,8	77,78 ± 1,89	85,45 ± 0,64	72,73 ± 1,87
900	85,67 ± 0,75	84,07 ± 1,59	82,2 ± 0,7	81,23 ± 0,8	86,04 ± 0,71	77,18 ± 2,01	85,23 ± 0,73	72,65 ± 1,21
1000	85,6 ± 0,8	84,53 ± 1,36	82,33 ± 0,6	81,57 ± 0,76	86,29 ± 0,86	78,27 ± 1,78	85,22 ± 0,89	72,33 ± 1,62

Fonte: Elaborada pelo autor.

Uma conclusão importante que pode ser visualizada na Figura 17 é que ao se atingir uma certa quantidade de perguntas, o algoritmo alcança um alto nível de estabilização em termos de *GS*. Ao se alcançar esse limite de avaliações, torna-se desnecessária a exigência de mais interações com o usuário. Por exemplo, analisando especificamente a I\_100 na Figura 17a, verifica-se que, utilizando o LMS, atinge-se um *GS* superior a 80% a partir de 100 perguntas. Para a mesma instância, porém na Figura 17b com o MLP, esse valor de estabilização seria por volta de 200 perguntas no qual alcança-se um *GS* de superior a 70%.

Logicamente, essa estabilização é intimamente associada a técnica de aprendizado de máquina e ao tamanho da instância empregada, afinal, é natural que quanto maior a solução,

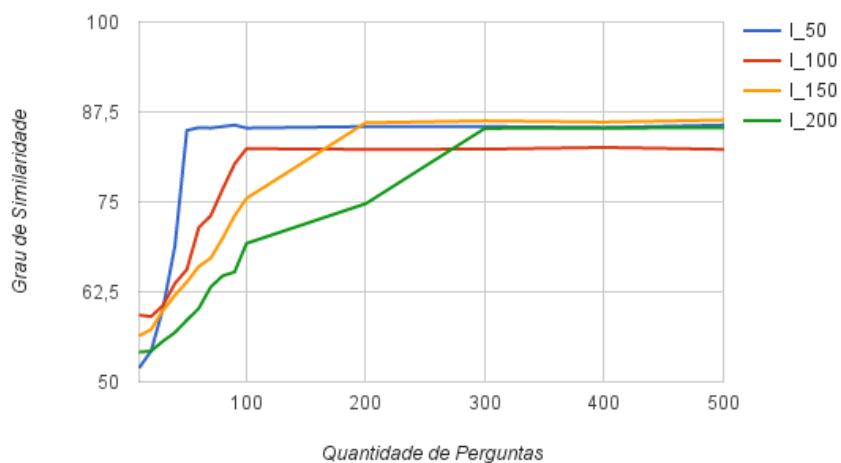
maior será a dificuldade em se aproximar das preferências do usuário. Ressalta-se a importância desse valor de estabilização pois o mesmo será aplicado nas próximas métricas como quantidade de perguntas realizadas. Exemplificando, para a I\_50 utilizando LMS, os testes referentes ao *Fator de Similaridade* e *Preço da Preferência* serão executados com uma quantidade fixa de 60 perguntas, haja vista que a partir desse valor ocorre a estabilização do *GS*. No Quadro 2 é possível se visualizar o restante dos valores para as demais instâncias e algoritmos de aprendizado.

Quadro 2 – Relação entre Quantidade de Perguntas e Instâncias Utilizadas

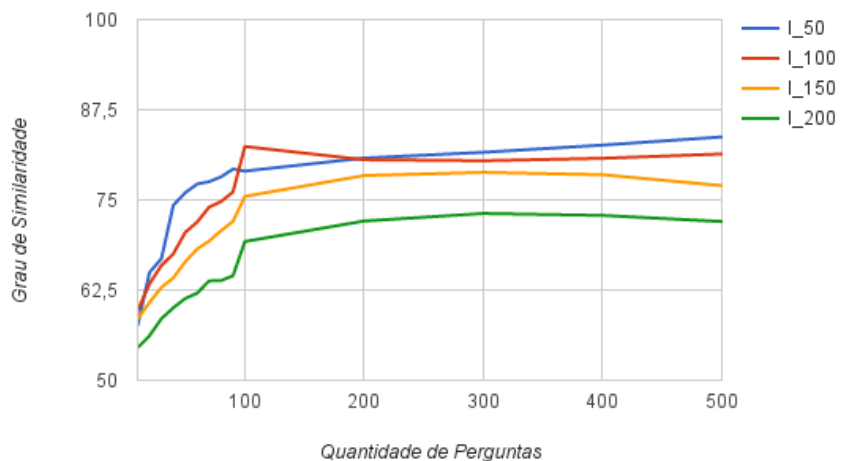
Instâncias	I_50		I_100		I_150		I_200	
Técnica de Aprendizado	LMS	MLP	LMS	MLP	LMS	MLP	LMS	MLP
Quantidade de Perguntas	60	200	100	200	200	200	300	200

Fonte: Elaborado pelo autor.

Figura 17 – Relação entre *Grau de Similaridade* e Quantidade de Perguntas



(a) Técnica de Aprendizado: LMS.



(b) Técnica de Aprendizado: MLP.

Fonte: Elaborada pelo autor.

Logo, através dessa análise constatou-se que o valor referente ao *GS* tem uma intrínseca relação com a quantidade de perguntas avaliadas pelo ser humano. Por sua vez, estas conclusões sugerem que a proposta apresentada passa pelo teste de sanidade ao demonstrar que de fato as preferências do usuário são incorporadas nas soluções, respondendo assim, a  $QP_1$ .

#### 5.4.2 Ganho Subjetivo

Complementando o objetivo anterior, neste tópico será analisado de forma mais efetiva o quanto é ganho na métrica *Fator de Similaridade (FS)* à medida que se amplia o peso  $\beta$  referente às avaliações humanas na função objetivo. Dessa forma, objetiva-se responder a seguinte questão de pesquisa:

**$QP_2$ : Quanto se ganha em preferências incorporadas quando se amplia a influência humana?**

A fim de se mostrar tal comportamento foram realizados testes nos quais fixou-se o peso  $\alpha$  da função *score* em 1 e o peso  $\beta$  da função *she* variou-se de 0 a 1 com intervalos uniformes de 0,1. Dessa forma verificou-se gradativamente o comportamento da proposta desde uma configuração totalmente sem a participação humana ( $\alpha = 1$  e  $\beta = 0$ ) até uma opção cujo peso da influência humana é equivalente ao peso do *score* ( $\alpha = 1$  e  $\beta = 1$ ).

A Tabela 2 apresenta detalhadamente os valores referentes ao *GS* e *FS* para diferentes configurações no peso  $\beta$  utilizando LMS e MLP. Analisando a I\_50 com uma configuração  $\beta = 0,5$ , constata-se que utilizando as técnicas LMS e MLP atinge-se respectivamente um *FS* de 33,6% e 30,2%. Quando se dobra esse peso para  $\beta = 1$ , os valores crescem consideravelmente para 57,2% e 48,7%. Em termos gerais, quando comparam-se as soluções geradas sem influência da avaliação humana ( $\alpha = 1$  e  $\beta = 0$ ) com as soluções que consideram equilibradamente os pesos ( $\alpha = 1$  e  $\beta = 1$ ), a média de ganho em *FS* para todas instâncias utilizando LMS é de 59,9%, enquanto com o MLP é de 40,37%. Isso significa um expressivo ganho de adequação às preferências do usuário, que evidentemente, é dependente do ajuste de peso dado ao aspecto subjetivo na função de avaliação.

A Figura 18 apresenta os valores de *GS* à medida que o peso  $\beta$  é aumentado para diferentes algoritmos de aprendizado. Logo, os resultados encontrados para as 4 instâncias são similares ao demonstrar que o valor de *FS* cresce proporcionalmente ao acréscimo do  $\beta$ , respondendo assim, a  $QP_2$ .

Tabela 2 – Valores referentes ao *Grau de Similaridade* e *Fator de Similaridade* ampliando o peso  $\beta$  utilizando LMS e MLP

$\beta$	I_50				I_100			
	LMS		MLP		LMS		MLP	
	GS	FS	GS	FS	GS	FS	GS	FS
<b>0</b>	54,27%	0,0%	54,33%	0,0%	59%	0,0%	58,37%	0,0%
<b>0,1</b>	57,6%	6,1%	58,33%	7,4%	63,6%	7,8%	63,4%	8,6%
<b>0,2</b>	59,73%	10,1%	59,33%	9,2%	67,17%	13,8%	65,9%	12,9%
<b>0,3</b>	63,73%	17,4%	62,73%	15,5%	70,1%	18,8%	68,47%	17,3%
<b>0,4</b>	68,67%	26,5%	66,87%	23,1%	73,13%	23,9%	70,97%	21,6%
<b>0,5</b>	72,53%	33,6%	70,73%	30,2%	75,93%	28,7%	71,8%	23,0%
<b>0,6</b>	76,33%	40,6%	72,67%	33,8%	79,77%	35,2%	73,73%	26,3%
<b>0,7</b>	80,53%	48,4%	76,47%	40,8%	81,03%	37,3%	75,37%	29,1%
<b>0,8</b>	82,8%	52,6%	77,67%	43,0%	81,37%	37,9%	78%	33,6%
<b>0,9</b>	83,73%	54,3%	79,4%	46,1%	82,03%	39,0%	79,33%	35,9%
<b>1</b>	85,33%	57,2%	80,8%	48,7%	82,4%	39,7%	80,57%	38,0%

$\beta$	I_150				I_200			
	LMS		MLP		LMS		MLP	
	GS	FS	GS	FS	GS	FS	GS	FS
<b>0</b>	56,11%	0,0%	56,07%	0,0%	54,03%	0,0%	53,37%	0,0%
<b>0,1</b>	61,78%	10,1%	60,78%	8,4%	58,42	8,1%	57,38%	7,5%
<b>0,2</b>	66,96%	19,3%	64,22%	14,5%	63,12%	16,8%	60,47%	13,3%
<b>0,3</b>	71,13%	26,8%	67,02%	19,5%	67,7%	25,3%	63,35%	18,7%
<b>0,4</b>	75,27%	34,1%	69,27%	23,5%	72,17	33,6%	65,02%	21,8%
<b>0,5</b>	79,22%	41,2%	71,49%	27,5%	75,68%	40,1%	66,58%	24,8%
<b>0,6</b>	82,62%	47,2%	73,27%	30,7%	79,17	46,5%	67,77%	27,0%
<b>0,7</b>	83,49%	48,8%	74,89%	33,6%	81,55%	50,9%	68,97%	29,2%
<b>0,8</b>	84,53%	50,7%	76,67%	36,7%	83,2%	54,0%	69,48%	30,2%
<b>0,9</b>	85,27%	52,0%	77,38%	38,0%	84,28%	56,0%	70,73%	32,5%
<b>1</b>	86%	53,3%	78,38%	39,8%	85,22%	57,7%	72,07%	35,0%

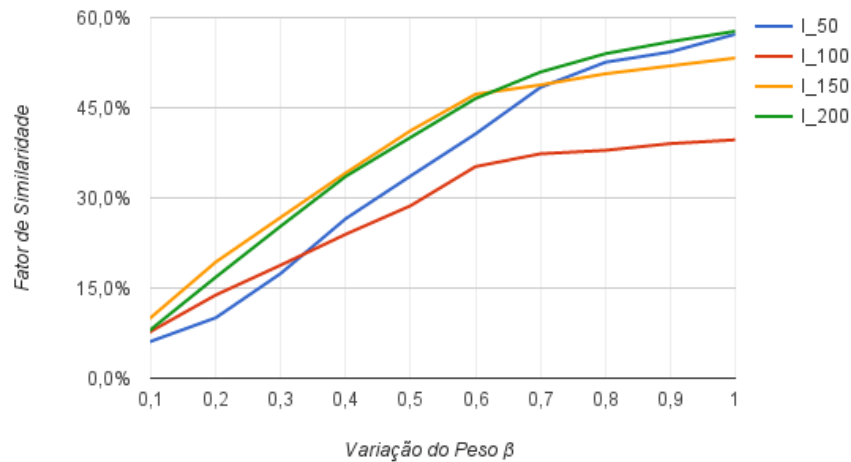
Fonte: Elaborada pelo autor.

### 5.4.3 Perda Objetiva

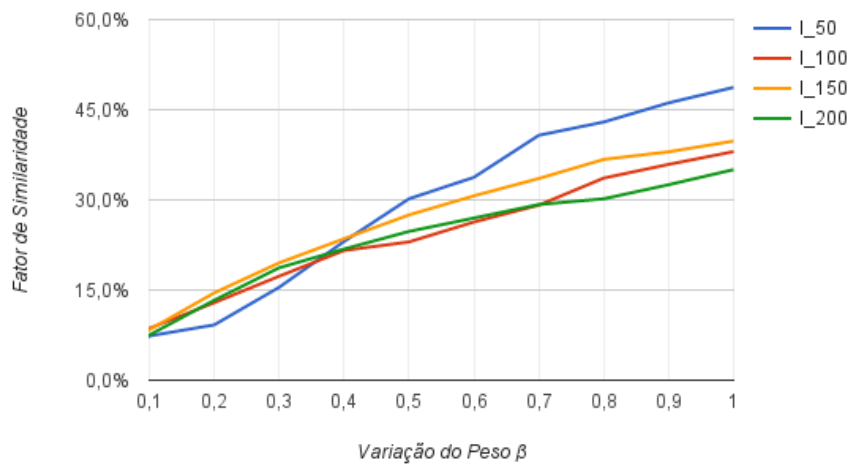
No tópico anterior demonstrou-se o aumento do *FS* em função do peso dado às preferências do usuário. Neste, através da métrica *Preço da Preferência (PP)* será esclarecido o impacto de tal participação humana sobre o valor de *score* das soluções geradas pela abordagem. Afinal, é plausível que alguns dos requisitos preferíveis pelo usuário não sejam bons do ponto de vista objetivo, culminando assim, em uma natural perda de *score*. Dessa forma, visa-se responder



Figura 18 – Fator de Similaridade



(a) Técnica de Aprendizizado: LMS.



(b) Técnica de Aprendizizado: MLP.

Fonte: Elaborada pelo autor.

a questão de pesquisa definida abaixo:

***QP*<sub>3</sub>: Qual a perda objetiva para se garantir uma solução mais adequada em termos subjetivos?**

Tal qual utilizado anteriormente, para responder essa pergunta de pesquisa também foram realizados testes nos quais fixou-se o peso  $\alpha$  da função *score* em 1 e o peso  $\beta$  da função *she* variou-se de 0 a 1 com intervalos uniformes de 0,1. A Tabela 3 apresenta os valores de *score* e *PP* para diferentes configurações no peso  $\beta$  utilizando LMS e MLP. Analisando especificamente a I\_150 com uma configuração  $\beta = 0,5$ , pode-se verificar que a perda em *score* utilizando a técnica LMS é de 5,3%, enquanto com MLP é de 2,3%. Dobrando esse peso para  $\beta = 1$ , atinge-se respectivamente 10,1% e 5,2%. Sob um ponto de vista geral, os dados demonstram

que para todas as instâncias utilizando o LMS, a perda média do valor de *score* foi de 13,34% quando comparadas as soluções geradas sem influência da avaliação humana ( $\alpha = 1$  e  $\beta = 0$ ) com as soluções geradas considerando pesos iguais para as funções *score* e *she* ( $\alpha = 1$  e  $\beta = 1$ ). No caso do MLP a queda média foi de 7,02%.

Tabela 3 – Valores referentes ao *score* e *Preço da Preferência* ampliando o peso  $\beta$  utilizando LMS e MLP

$\beta$	I_50				I_100			
	LMS		MLP		LMS		MLP	
	<i>score</i>	PP	<i>score</i>	PP	<i>score</i>	PP	<i>score</i>	PP
<b>0</b>	115,87	0,0%	115,79	0,0%	229,51	0%	229,32	0,0%
<b>0,1</b>	115,69	0,16%	115,61	0,16%	229,13	0,17%	229,39	0,03%
<b>0,2</b>	115,21	0,57%	115,38	0,36%	227,59	0,84%	228,33	0,43%
<b>0,3</b>	113,77	1,85%	114,47	1,15%	225,46	1,17%	226,66	1,16%
<b>0,4</b>	111,8	3,64%	112,61	2,82%	222,95	2,94%	224,87	1,98%
<b>0,5</b>	109,57	5,75%	110,93	4,38%	220,11	4,27%	224,16	2,30%
<b>0,6</b>	107,3	7,99%	109,79	5,46%	215,49	6,51%	222,38	3,12%
<b>0,7</b>	104,41	10,98%	107,35	7,86%	213,71	7,34%	220,81	3,85%
<b>0,8</b>	102,52	13,02%	106,71	8,51%	212,94	7,78%	217,7	5,35%
<b>0,9</b>	101,29	14,39%	105,29	9,97%	211,57	8,48%	216,33	6,00%
<b>1</b>	99,89	16,00%	104,09	11,24%	210,9	8,82%	214,54	6,89%

$\beta$	I_150				I_200			
	LMS		MLP		LMS		MLP	
	<i>score</i>	PP	<i>score</i>	PP	<i>score</i>	PP	<i>score</i>	PP
<b>0</b>	320,82	0%	320,72	0,0%	458,53	0%	458,38	0,0%
<b>0,1</b>	319,85	0,03%	320,35	0,12%	452,99	1,22%	457,66	0,16%
<b>0,2</b>	317,49	1,05%	319,31	0,44%	448,25	2,29%	455,64	0,60%
<b>0,3</b>	314,03	2,16%	317,15	1,13%	437,23	4,87%	453,36	1,11%
<b>0,4</b>	309,34	3,71%	315,36	1,70%	428,78	6,95%	451,42	1,54%
<b>0,5</b>	303,75	5,62%	313,42	2,33%	416,05	10,21%	449,09	2,07%
<b>0,6</b>	298,04	7,64%	311,32	3,02%	416,66	10,05%	447,42	2,45%
<b>0,7</b>	296,1	8,35%	309,29	3,70%	404,92	13,24%	444,84	3,04%
<b>0,8</b>	293,35	9,36%	306,74	4,56%	402,85	13,82%	444,03	3,23%
<b>0,9</b>	291,17	10,18%	305,23	5,07%	395,18	16,03%	441,74	3,77%
<b>1</b>	288,31	11,28%	304,05	5,48%	391,01	17,27%	438,73	4,48%

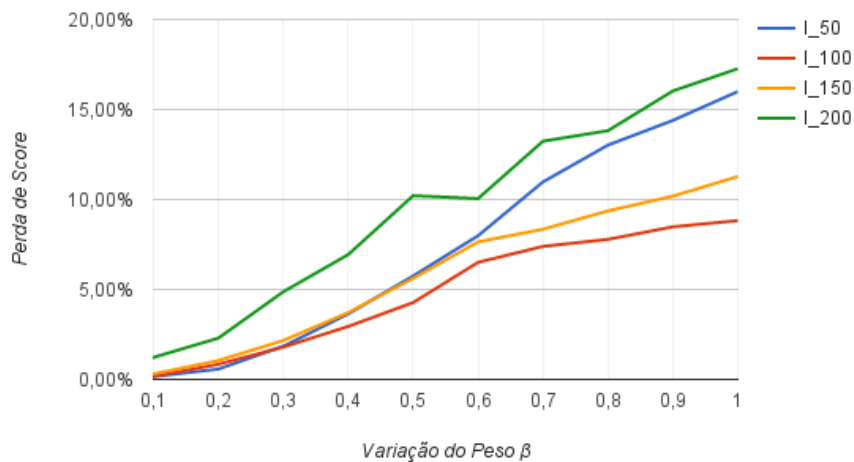
Fonte: Elaborada pelo autor.

É importante ressaltar que o fato dos resultados referentes à aplicação do modelo de aprendizado com MLP serem menores do que os do LMS, é consequência do *FS* atingido por cada técnica de aprendizado de máquina, conforme já constatado na Tabela 2. Como o LMS

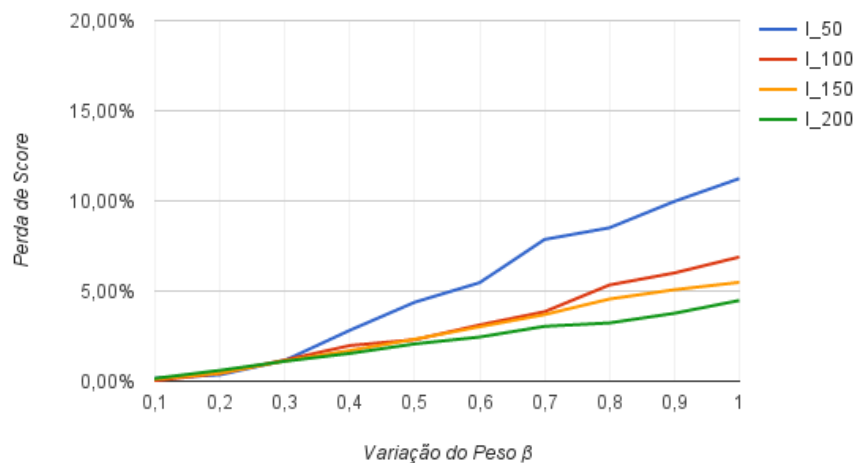
consegue um alto *FS*, implicitamente a perda com relação ao *score* será maior.

A Figura 19 proporciona uma visão geral sobre o comportamento do *PP* à medida que o peso da opinião humana é incrementado. Observa-se que as soluções encontradas para todas as instâncias com cada uma das técnicas de aprendizado, apresentam similar tendência de decréscimo no valor da função *score*. Assim, constata-se o fato de que incorporar as preferências do usuário implica em naturalmente perder um pouco em termos objetivos, ou seja, em *score*. Todavia, dado o substancial ganho em *FS* comprovado anteriormente, uma pequena perda no valor de *score* de uma solução não significa necessariamente perda de qualidade da solução final. Consequentemente, tal análise proporciona responder a *QP*<sub>3</sub>.

Figura 19 – Preço da Preferência



(a) Técnica de Aprendizado: LMS.



(b) Técnica de Aprendizado: MLP.

Fonte: Elaborada pelo autor.

#### 5.4.4 Trade-off entre Ganho Subjetivo e a Perda Objetiva

A partir das análises anteriores obtêm-se duas importantes constatações. A primeira é com relação a alta quantidade de requisitos similares quando se compara uma solução candidata e o indivíduo-alvo, ou seja, elevado Grau de Similaridade (*GS*). A segunda é sobre o quanto é perdido em termos de *score* devido a inclusão das preferências humanas, mensurado pela métrica (*PP*). Dito isso, almeja-se responder a questão de pesquisa descrita a seguir:

***QP*<sub>4</sub>: Qual a relação entre o quanto se ganha em preferências incluídas e, consequentemente, quanto se perde em termos objetivos?**

A Figura 20 demonstra de maneira mais evidente a relação entre as métricas *GS* e *PP*, destacando a pouca perda de *score* necessária para se atingir um alto *GS*. À direita vê-se o quanto foi atingido em *GS* e à esquerda o quanto foi perdido em *score* para todas as instâncias. Analisando por exemplo a instância I\_100, verifica-se que utilizando o LMS perdeu-se apenas 8,10% em *score* para se atingir um *GS* de 82,40%. Aplicando o MLP, a perda foi 6,40% chegando-se a 80,60% de *GS*.

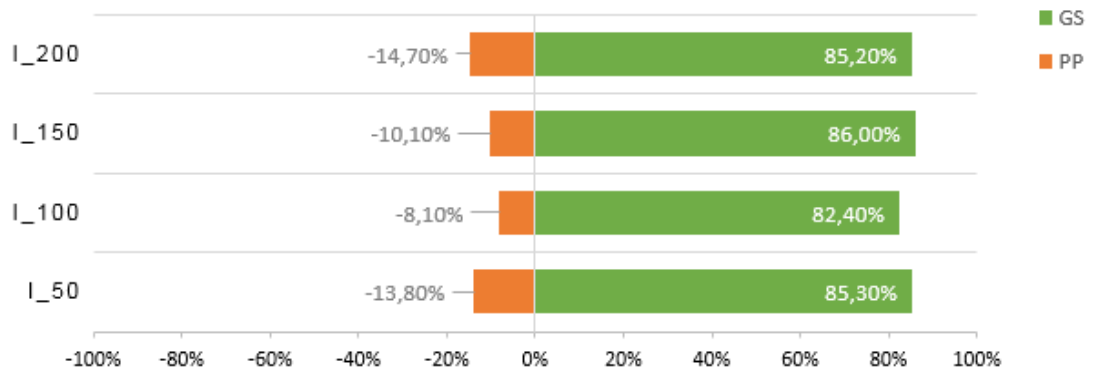
Em síntese, utilizando LMS, a perda média de *score* para todas instâncias foi de 11,6%, e atingiu-se em média um *GS* de 84,7%. Adotando o MLP, os valores médios obtidos foram 6,5% e 77,9% para perda de *score* e *GS* atingido, respectivamente. Tais resultados contribuem na resposta da *QP*<sub>4</sub> ao se comprovar que, além de incorporar as preferências humanas, a perda em *score* é relativamente pequena. Portanto, constata-se que a abordagem de otimização interativa proposta, aliada à formulação iNRP, é capaz de produzir bons resultados do ponto de vista objetivo, com o benefício de que a maioria das preferências do usuário são incorporadas à solução final.

#### 5.5 AMEAÇAS À VALIDADE

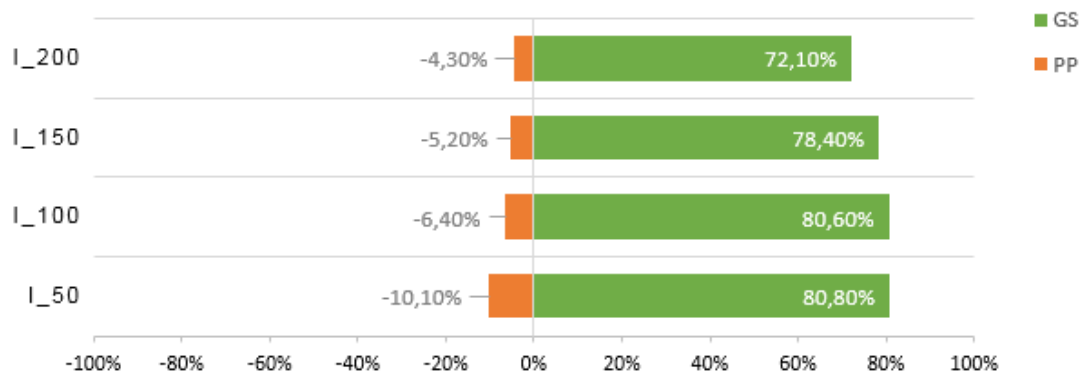
O trabalho de Barros e Dias-Neto (2011) apresenta uma lista com diversas ameaças à validade de experimentos em SBSE, que abrangem desde o projeto até a execução de estudos empíricos. Tais ameaças são classificadas como: Ameaças à Validade de Conclusões, Ameaças à Validade Interna, Ameaças à Validade de Construção e Ameaça à Validade Externa.

Nesta seção serão discutidos os riscos à validade que podem afetar os experimentos realizados neste trabalho, além de informado o que eventualmente foi desenvolvido para atenuar

Figura 20 – Relação entre Grau de Similaridade e Preço da Preferência



(a) Técnica de Aprendizado: LMS.



(b) Técnica de Aprendizado: MLP.

Fonte: Elaborada pelo autor.

os efeitos de cada risco.

#### a) Ameaças à Validade de Conclusões

- Variação aleatória: devido a característica estocástica na geração de soluções oriundas da metaheurística utilizadas neste trabalho, cada execução do algoritmo proposto pode gerar resultados diferentes. Por isso, durante os experimentos, para cada uma das configurações dos pesos da função objetivo, quantidade de interações com o usuário, instâncias e técnicas de aprendizado de máquina, foram realizadas 30 execuções considerando-se a média e os desvios padrões. Em (ARCURI; BRIAND, 2014) ressalta-se que esse é um número de execuções aceitável para experimentos em SBSE, no entanto, uma quantidade maior de execuções poderia oferecer maior precisão dos resultados.
- Falta de boa estatística descritiva: apesar de não terem sido realizados testes estatísticos, a maioria das tabelas de resultado apresentam a média e o desvio padrão de todas as execuções com uma determinada configuração.

- Falta de uma significativa linha base de comparação: o Problema do Próximo *Release* é um problema bastante explorado em SBSE, porém, não existem outras abordagens que resolvam através de um Algoritmo Genético Iterativo. Logo, os resultados foram comparados apenas com os resultados automáticos, ou seja, sem intervenção subjetiva.

#### b) Ameaças à Validade Interna

- Parametrização: na seção referente às configurações são apresentados os parâmetros utilizados para o IGA, cujo valores foram obtidos empiricamente. Com relação aos parâmetros relacionados às técnicas de aprendizado de máquina optou-se pelos valores sugeridos pela ferramenta *WEKA*. Uma configuração de parâmetros mais refinada, principalmente com relação às técnicas de aprendizado, poderia proporcionar melhores resultados.
- Falta de discussão sobre a codificação: neste trabalho não são realizadas discussões acerca dos detalhes de implementação usada no estudo empírico, porém, o código fonte pode ser encontrado na página *web* de suporte desta pesquisa (<http://goes.uece.br/allyssonaraujo/architecture4inrp/>) e nos Apêndices A e B desta dissertação.
- Falta de clareza das ferramentas e procedimento para obtenção de instâncias: como todas instâncias utilizadas no trabalho são artificiais, possibilitou-se esclarecer toda a composição relacionadas às mesmas na Seção 5.1.

#### c) Ameaças à Validade de Construção

- Falta de avaliação de custo: este trabalho não realiza um estudo sobre custo de execução ou performance. Não obstante, a Seção 5.4 apresenta uma discussão relacionada a quantidade de interações necessárias para estabilização de cada técnica de aprendizado de máquina utilizada nos experimentos.
- Falta de avaliação na validade das medidas de eficácia: conforme apresentado na Seção 3.1, vários estudos exploram o NRP a partir de diversas formulações como, por exemplo, em (BAKER et al., 2006) e (BAGNALL et al., 2001). Neste trabalho, porém, utilizou-se uma generalização da função de avaliação modelada tradicionalmente. A motivação, o impacto e as métricas propostas para avaliar tal adaptação são amplamente discutidos ao longo do trabalho.
- Falta de discussão sobre o modelo submetido a otimização: a modelagem do

NRP utilizada neste trabalho é uma simplificação da realidade do contexto de desenvolvimento de software. Visto que a abstração de peculiaridades como precedência de requisitos, por exemplo, limita uma maior aproximação do mundo real.

**d) Ameaças à Validade Externa**

- Falta de avaliação sobre escalabilidade: nos experimentos deste trabalho, foram utilizadas instâncias com 50, 100, 150 e 200 requisitos. Tais instâncias foram suficientes para analisar o comportamento da arquitetura proposta, porém, testes com mais requisitos proporcionariam maior generalidade dos resultados.

## 5.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo discutiu-se o estudo empírico realizado em prol da validação e avaliação da arquitetura proposta. A princípio elucidou-se as instâncias, configurações e métricas desenvolvidas. Em seguida analisou-se o comportamento dos resultados visando responder quatro questões de pesquisa.

Verificou-se a relação entre a quantidade de perguntas subjetivas e a adequação das preferências humanas. Como resultado, mostrou-se que o ganho em *GS* foi substancial, sugerindo assim, que a proposta passa pelo teste de sanidade ao incorporar de fato as preferências do usuário nas soluções.

Após isso, considerou-se relevante averiguar através da métrica *FS* a diferença percentualmente em *GS* à medida que o peso da avaliação humana na função objetivo é incrementado. Tal análise possibilita se assimilar a real influência que o peso  $\beta$  exerce sobre os ganhos com relação a inclusão do conhecimento humano na solução final.

Todavia, verificou-se que é natural haver perda em termos de *score* na solução, haja vista que muitas vezes os requisitos preferíveis pelo usuário podem não ser bons em termos objetivos. Dito isso, por intermédio do *PP* apurou-se quanto se perde em *score* para se obter uma solução mais adequada do ponto de vista subjetivo.

Posteriormente, uma importante análise é realizada ao se sintetizar o quanto se perde em *score* para se alcançar um *GS* significativo, ou seja, o *trade-off* entre perda objetiva e ganho subjetivo. Logo, comprovou-se empiricamente que os resultados conseguem incorporar devidamente a opinião humana e ainda se manter bons com relação ao aspecto objetivo específico do problema.

Essa relação perde/ganha do *score* pode ser adaptada através da configuração dos pesos na função objetivo, ou seja, caso o usuário prefira uma solução que privilegie mais o aspecto objetivo basta adequar o peso  $\alpha$  em detrimento do  $\beta$ . Assim, cabe ao especialista do domínio identificar o cenário no qual está inserido e adequar a arquitetura da forma que melhor corresponda à sua realidade.

Por fim discute-se as ameaças à validade que podem afetar os experimentos realizados neste trabalho. Afinal, considera-se relevante analisar todo o processo que envolve a execução do estudo empírico e, conseqüentemente, demonstrar as medidas adotadas para atenuar os efeitos de cada risco.

Portanto, demonstrou-se neste capítulo a implementação da arquitetura proposta para resolução do Problema do Próximo *Release* através de uma modelagem interativa. A partir do estudo empírico realizado, foi possível atingir os dois últimos objetivos específicos elencados no Capítulo 1: “**b**” e “**c**”.



## 6 CONSIDERAÇÕES FINAIS

A seleção de requisitos para o próximo *release* é uma tarefa inerentemente complexa no desenvolvimento de software iterativo e incremental, por principalmente englobar aspectos muito relevantes para o resultado de um projeto, como a presença e a subjetividade humana. Todavia, quando apenas técnicas de busca tradicionais são usadas para atacar tais problemas, possivelmente as preferências do usuário não são incisivamente influentes ao longo do processo.

A Otimização Interativa é uma linha de pesquisa que utiliza o ser humano de forma protagonista no processo de busca por soluções. Recomenda-se sua utilização da mesma principalmente quando a opinião humana pode contribuir efetivamente para resultados mais adequados ou quando a função de avaliação não pode ser capturada através de modelos matemáticos puramente objetivos.

Usufruindo dos benefícios inerentes à Otimização Interativa e SBSE, o objetivo deste trabalho foi desenvolver um arquitetura factível que consiga incorporar as preferências do usuário, mas que também saiba lidar com a fadiga humana. Sua concepção é fundamentada em três componentes distintos que, de forma complementar, conseguem se comunicar e proporcionar ao processo de busca ser guiado pela opinião humana. Para abrandar a fadiga, optou-se pelo uso de técnicas de aprendizado de máquina capazes de modelar as preferências do usuário e substituí-lo posteriormente, ou seja, evita-se a necessidade de se solicitar uma avaliação subjetiva para absolutamente todas as soluções. Dentre as possibilidades proporcionadas pela abordagem pode-se mencionar a configuração dos pesos, proporcionando ao usuário a oportunidade de adequar a arquitetura devidamente às necessidades e ao cenário confrontado.

Como pode-se constatar, o estudo de caso explorado nesta pesquisa foi uma abordagem interativa do Problema do Próximo *Release*, denominado como iNRP. Após se definir uma formulação interativa para o mesmo, realizou-se uma avaliação experimental na qual permitiu observar que de fato os aspectos subjetivos são incorporados na solução final. Conseqüentemente, é natural que haja uma perda em termos de *score*, porém, constatou-se que essa perda pode ser considerada relativamente pequena dado o substancial acréscimo em relação à quantidade de preferências atendidas. Em termos de resultados, através da resposta referente a  $QP_4$ , foi possível verificar que, em síntese, para todas as instâncias atingiu-se em média um *GS* de 84,7%, para uma perda de *score* em 11,6%.

Todavia, ainda é plausível afirmar que a arquitetura proposta é genérica o suficiente para ser aplicada em diversos outros problemas tratados pela SBSE como, por exemplo, seleção

de casos testes, *features* para uma Linha de Produtos de Software, entre outros. Portanto, para sua aplicação bastaria modelar o problema a ser explorado de acordo com os requisitos definidos pela arquitetura, tal qual foi modelado do NRP para o iNRP.

## 6.1 CONTRIBUIÇÕES

Visando consolidar as ideias absorvidas, lista-se a seguir as principais contribuições advindas da presente pesquisa:

- a) **Proposta de arquitetura:** definiu-se um conjunto de componentes e fases que, em conjunto, proporcionam ao usuário a possibilidade de incorporar suas preferências sem a necessidade de se obter uma avaliação subjetiva em todas as avaliações do algoritmo. Dada sua natureza genérica, é válido afirmar que a proposta pode ser aplicada em outros problemas explorados pela SBSE.
- b) **Formulação interativa do NRP:** tornou-se necessário adequar o problema explorado aos requisitos da arquitetura, haja vista que o mesmo em sua versão canônica não inclui incisivamente o usuário. Dessa forma, exemplificou-se através da formulação interativa do NRP como pode ocorrer a etapa de modelagem de um problema para que seja resolvido através da arquitetura proposta.
- c) **Elaboração de métricas:** para uma análise empírica mais profunda necessitou-se elaborar algumas métricas aptas a exprimir determinado comportamento da abordagem. Dito isso, sugere-se que as métricas geradas nesse trabalho são genéricas o suficientes para serem explorados em outros contextos que se utilizem dos conceitos de Otimização Interativa em SBSE.
- d) **Implementação do algoritmo:** implementou-se um Algoritmo Genético Interativo que consegue capturar as avaliações subjetivas do usuário durante o cálculo de aptidão dos indivíduos. Entretanto, outro diferencial foi a comunicação com o Modelo de Aprendizado que, após aprender o perfil de avaliação do usuário, consegue simular suas preferências diante de uma determinada seleção de requisitos.
- e) **Análise dos trabalhos de Otimização Interativa em SBSE:** elaborou-se uma breve análise bibliográfica sobre os trabalhos categorizados em tal contexto. Também apresentou-se algumas ideias e conjecturas que podem ser relevantes para futuros trabalhos.

## 6.2 LIMITAÇÕES DA PESQUISA

Acredita-se que todos os objetivos traçados por esta pesquisa foram devidamente atingidos ao longo do seu desenvolvimento. Entretanto, algumas limitações podem ser constatadas.

São elas:

- a) A ausência de experimentos com instâncias reais, seres humanos ou em um projeto real de desenvolvimento de software que possibilite uma análise mais ampla;
- b) A supressão das interdependências funcionais e de valor entre os requisitos;
- c) A escassez de comparação com outras abordagens interativas que possibilite obter resultados mais maduros;
- d) A falta de testes com mais técnicas de aprendizado de máquina que possam contribuir para uma validação mais robusta;
- e) A ausência de uma análise detalhada em termos de desempenho computacional;
- f) A carência na diversificação da parametrização utilizada nos técnicas de aprendizado de máquina.

## 6.3 TRABALHOS FUTUROS

O aperfeiçoamento e a continuidade deste trabalho, incluindo as próprias limitações apresentadas anteriormente, constituem as oportunidades para trabalhos futuros, que incluem:

- a) Verificar estratégias mais elegantes de se mensurar a quantidade de perguntas a ser direcionada ao usuário. Por exemplo, solicitar a priori ao usuário quanto o mesmo está disposto a ganhar em *GS* ou perder em *score* e, a partir de uma execução prévia do algoritmo com o simulador, obter a configuração de pesos e quantidade de perguntas ideais a serem utilizadas;
- b) Inserir interdependências entre requisitos, incluindo as funcionais e de valor, tornando a abordagem mais condizente para ser aplicável num contexto real de desenvolvimento de software;
- c) Desenvolver uma ferramenta com interface para aplicação ambiente real, possibilitando a realização de um estudo empírico mais completo, por exemplo, com a participação de seres humanos;
- d) Formular outros problemas sob um viés interativo para serem aplicados na

arquitetura como, por exemplo, seleção de casos de teste ou *features* em uma Linha de Produto de Software.

- e) Testar novas formas de se atribuir uma *she* para as soluções como, por exemplo, uma avaliação *fuzzyficada*;
- f) Incrementar a arquitetura com a utilização de outras metaheurística interativas ou multiobjetivas, além de possibilitar o uso de mais técnicas de aprendizado de máquina;
- g) Incluir um maior refinamento no tratamento do nível de confiança referente às técnicas de aprendizado de máquina;
- h) Expandir os experimentos para novas configurações, incluindo uma maior variação na restrição de *budget*.

## REFERÊNCIAS

- AKERKAR, R.; SAJJA, P. **Knowledge-based systems**. [S.l.]: Jones & Bartlett Publishers, 2010.
- AKKER, M. van den; BRINKKEMPER, S.; DIEPEN, G.; VERSENDAAAL, J. Determination of the next release of a software product: an approach using integer linear programming. In: **CAiSE Short Paper Proceedings**. [S.l.: s.n.], 2005.
- ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2014.
- ARCURI, A.; BRIAND, L. A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. **Software Testing, Verification and Reliability**, Wiley Online Library, v. 24, n. 3, p. 219–250, 2014.
- BAGNALL, A. J.; RAYWARD-SMITH, V. J.; WHITTLEY, I. M. The next release problem. **Information and Software Technology**, Elsevier, v. 43, n. 14, p. 883–890, 2001.
- BAKER, P.; HARMAN, M.; STEINHOFEL, K.; SKALIOTIS, A. Search based approaches to component selection and prioritization for the next release problem. In: **IEEE. Software Maintenance, 2006. ICSM'06. 22nd IEEE International Conference on**. [S.l.], 2006. p. 176–185.
- BARRETO, J. M. Inteligência artificial no limiar do século xxi. **Florianópolis: PPP edições**, v. 97, 1999.
- BARROS, M. d. O.; DIAS-NETO, A. C. A survey of empirical investigations on ssbse papers. In: **Search Based Software Engineering**. [S.l.]: Springer, 2011. p. 268–268.
- BAVOTA, G.; CARNEVALE, F.; LUCIA, A. D.; PENTA, M. D.; OLIVETO, R. Putting the developer in-the-loop: an interactive ga for software re-modularization. In: **Search Based Software Engineering**. [S.l.]: Springer, 2012. p. 75–89.
- BILES, J. Genjam: A genetic algorithm for generating jazz solos. In: **INTERNATIONAL COMPUTER MUSIC ACCOCIATION. Proceedings of the International Computer Music Conference**. [S.l.], 1994. p. 131–131.
- BOSCHETTI, F.; MORESI, L. Interactive inversion in geosciences. **Geophysics**, Society of Exploration Geophysicists, v. 66, n. 4, p. 1226–1234, 2001.
- BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: Livros Técnicos e Científicos, 2000.
- CALDWELL, C.; JOHNSTON, V. S. Tracking a criminal suspect through "face-space" with a genetic algorithm. In: **ICGA**. [S.l.: s.n.], 1991. v. 4, p. 416–421.
- CARVALHO, L. A. V. **Data Mining: A mineração de dados no Marketing, Medicina, Economia, Engenharia e Administração**. São Paulo: Ed. Érica, 2002.
- FERREIRA, T. d. N.; SOUZA, J. T. d. An aco approach for the next release problem with dependency among requirements. In: **Proceedings of the 3rd Brazilian Workshop on Search-Based Software Engineering (WESB '12)**. Natal, RN, Brazil: [s.n.], 2012.
- GHANNEM, A.; BOUSSAIDI, G. E.; KESSENTINI, M. Model refactoring using interactive genetic algorithm. In: **Search Based Software Engineering**. [S.l.]: Springer, 2013. p. 96–110.

GREER, D.; RUHE, G. Software release planning: an evolutionary and iterative approach. **Information and Software Technology**, Elsevier, v. 46, n. 4, p. 243–253, 2004.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: an update. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, 2009. ISSN 1931-0145. Disponível em: <<http://dx.doi.org/10.1145/1656274.1656278>>.

HARMAN, M. Search based software engineering for program comprehension. In: IEEE. **Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on**. [S.l.], 2007. p. 3–13.

HARMAN, M.; CLARK, J. Metrics are fitness functions too. In: IEEE. **Software Metrics, 2004. Proceedings. 10th International Symposium on**. [S.l.], 2004. p. 58–69.

HARMAN, M.; JONES, B. F. Search-based software engineering. **Information and Software Technology**, Elsevier, v. 43, n. 14, p. 833–839, 2001.

HARMAN, M.; MANSOURI, S. A.; ZHANG, Y. Search based software engineering: A comprehensive analysis and review of trends techniques and applications. **Department of Computer Science, King's College London, Tech. Rep. TR-09-03**, 2009.

HARMAN, M.; MCMINN, P.; SOUZA, J. T. de; YOO, S. Search based software engineering: Techniques, taxonomy, tutorial. In: **Empirical Software Engineering and Verification**. [S.l.]: Springer, 2012. p. 1–59.

HAYKIN, S. S. **Redes neurais**. [S.l.]: Bookman, 2001.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: USA: University of Michigan, 1975.

HSU, F.-C.; HUANG, P. Providing an appropriate search space to solve the fatigue problem in interactive evolutionary computation. **New Generation Computing**, Springer, v. 23, n. 2, p. 115–127, 2005.

JIANG, H.; ZHANG, J.; XUAN, J.; REN, Z.; HU, Y. A hybrid aco algorithm for the next release problem. In: IEEE. **Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on**. [S.l.], 2010. p. 166–171.

KAMALIAN, R.; YEH, E.; ZHANG, Y.; AGOGINO, A. M.; TAKAGI, H. Reducing human fatigue in interactive evolutionary computation through fuzzy systems and machine learning systems. In: IEEE. **Fuzzy Systems, 2006 IEEE International Conference on**. [S.l.], 2006. p. 678–684.

KIM, H.-S.; CHO, S.-B. Application of interactive genetic algorithm to fashion design. **Engineering applications of artificial intelligence**, Elsevier, v. 13, n. 6, p. 635–644, 2000.

KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. [S.l.]: MIT press, 1992.

LACERDA, E. G. de; CARVALHO, A. de. Introdução aos algoritmos genéticos. **Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais**, v. 1, p. 99–148, 1999.

- LEWIS, M. A.; FAGG, A. H.; SOLIDUM, A. Genetic programming approach to the construction of a neural network for control of a walking robot. In: IEEE. **Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on.** [S.l.], 1992. p. 2618–2623.
- MAIA, C. L. B. **Uma Abordagem Integrada, Interativa e Multi-Objetiva para os Problemas de Seleção, Priorização e Alocação de Casos de Teste.** Dissertação (Mestrado) — MACC, Mestrado Acadêmico em Ciência da Computação, Universidade Estadual do Ceará, 2011.
- MARCULESCU, B.; FELDT, R.; TORKAR, R. A concept for an interactive search-based software testing system. In: **Search Based Software Engineering.** [S.l.]: Springer, 2012. p. 273–278.
- MARCULESCU, B.; FELDT, R.; TORKAR, R. Objective re-weighting to guide an interactive search based software testing system. In: IEEE. **Machine Learning and Applications (ICMLA), 2013 12th International Conference on.** [S.l.], 2013. v. 2, p. 102–107.
- MARCULESCU, B.; FELDT, R.; TORKAR, R.; POULDING, S. An initial industrial evaluation of interactive search-based testing for embedded software. **Applied Soft Computing,** Elsevier, 2014.
- MIETTINEN, K. **Nonlinear multiobjective optimization.** [S.l.]: Springer, 1999.
- MILLER, W.; SPOONER, D. L. Automatic generation of floating-point test data. **IEEE Transactions on Software Engineering,** IEEE, v. 2, n. 3, p. 223–226, 1976.
- MITCHELL, T. M. **Machine Learning.** 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.
- MONARD, M. C.; BARANAUSKAS, J. A. **Sistemas Inteligentes: Fundamentos e Aplicacoes.** [S.l.]: Ed. Manole, 2003.
- O'KEEFFE, M.; CINNÉIDE, M. Ó. Search-based refactoring for software maintenance. **Journal of Systems and Software,** Elsevier, v. 81, n. 4, p. 502–516, 2008.
- PACHECO, M. A. C. Algoritmos genéticos: princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro,** 1999.
- PARMEE, I.; HALL, A.; MILES, J.; NOYES, J.; SIMONS, C. et al. Discovery in design: Developing a people-centred computational approach. In: **DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia.** [S.l.: s.n.], 2006.
- PRESSMAN, R. S. **Engenharia de Software.** 6. ed. [S.l.]: McGraw-Hill, 2006.
- QU, R.; BURKE, E. K.; MCCOLLUM, B.; MERLOT, L. T.; LEE, S. Y. A survey of search methodologies and automated system development for examination timetabling. **Journal of scheduling,** Springer, v. 12, n. 1, p. 55–89, 2009.
- REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações.** [S.l.]: Editora Manole, 2003.
- ROTHERMEL, G.; UNTCH, R. H.; CHU, C.; HARROLD, M. J. Prioritizing test cases for regression testing. **Software Engineering, IEEE Transactions on,** IEEE, v. 27, n. 10, p. 929–948, 2001.

- ROUSSEEUW, P. J.; LEROY, A. M. **Robust regression and outlier detection**. [S.l.]: John Wiley & Sons, 2005.
- RUHE, G.; SALIU, M. O. The art and science of software release planning. **Software, IEEE, IEEE**, v. 22, n. 6, p. 47–53, 2005.
- SAGRADO, J. del; AGUILA, I. M. del; ORELLANA, F. J. Ant colony optimization for the next release problem: A comparative study. In: **Search Based Software Engineering (SSBSE), 2010 Second International Symposium on**. [S.l.: s.n.], 2010. p. 67–76.
- SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas curso prático. **Artliber**, SciELO Brasil, 2010.
- SIMONS, C. **Interactive evolutionary computing in early lifecycle software engineering design**. Tese (Doutorado) — University of the West of England, 2011.
- SIMONS, C.; SMITH, J. A comparison of meta-heuristic search for interactive software design. **Soft Computing**, Springer, v. 17, n. 11, p. 2147–2162, 2013.
- SIMONS, C. L.; PARMEE, I. C. Dynamic parameter control of interactive local search in uml software design. In: IEEE. **Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on**. [S.l.], 2010. p. 3397–3404.
- SIMONS, C. L.; PARMEE, I. C. Elegant object-oriented software design via interactive, evolutionary computation. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, IEEE, v. 42, n. 6, p. 1797–1805, 2012.
- SIMONS, C. L.; PARMEE, I. C.; GWYNLLYW, R. Interactive, evolutionary search in upstream object-oriented class design. **Software Engineering, IEEE Transactions on**, IEEE, v. 36, n. 6, p. 798–816, 2010.
- SIMONS, C. L.; SMITH, J.; WHITE, P. Interactive ant colony optimisation (iaco) for early lifecycle software design. **arXiv preprint arXiv:1212.5461**, 2012.
- SIMONS, C. L.; SMITH, J.; WHITE, P. Interactive ant colony optimization (iaco) for early lifecycle software design. **Swarm Intelligence**, Springer, v. 8, n. 2, p. 139–157, 2014.
- SIMS, K. **Artificial evolution for computer graphics**. [S.l.]: ACM, 1991.
- SOMMERVILLE, I. **Software Engineering**. [S.l.]: Addison Wesley, 2011.
- TAKAGI, H. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. **Proceedings of the IEEE**, IEEE, v. 89, n. 9, p. 1275–1296, 2001.
- TONELLA, P.; SUSI, A.; PALMA, F. Using interactive ga for requirements prioritization. In: IEEE. **Search Based Software Engineering (SSBSE), 2010 Second International Symposium on**. [S.l.], 2010. p. 57–66.
- TONELLA, P.; SUSI, A.; PALMA, F. Interactive requirements prioritization using a genetic algorithm. **Information and software technology**, Elsevier, v. 55, n. 1, p. 173–187, 2013.
- WANG, S.; WANG, X.; TAKAGI, H. User fatigue reduction by an absolute rating data-trained predictor in iec. In: IEEE. **Evolutionary Computation, 2006. CEC 2006. IEEE Congress on**. [S.l.], 2006. p. 2195–2200.



WEIMER, W.; FORREST, S.; GOUES, C. L.; NGUYEN, T. Automatic program repair with evolutionary computation. **Communications of the ACM**, ACM, v. 53, n. 5, p. 109–116, 2010.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2005.

XUAN, J.; JIANG, H.; REN, Z.; LUO, Z. Solving the large scale next release problem with a backbone-based multilevel algorithm. **Software Engineering, IEEE Transactions on**, IEEE, v. 38, n. 5, p. 1195–1212, 2012.

ZHANG, Y.; HARMAN, M.; MANSOURI, A. **Repository of Publications on Search Based Software Engineering**. 2011. <[http://crestweb.cs.ucl.ac.uk/resources/sbse\\_repository/](http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/)>. Acesso em: 18/11/2014.

## **APÊNDICES**

## APÊNDICE A – Resultados utilizando LMS

INSTÂNCIAS	ATRIBUTOS	10 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	54,93+/-2,72	53,67+/-3,19	53,33+/-2,94	53,2+/-3,64	53,07+/-3	53,27+/-3,44	51,87+/-3,79	53,4+/-4,07	52,93+/-3,82	53,73+/-3,53	51,93+/-3,33	5,46%
	Score	115,81+/-0,45	115,79+/-0,38	115,82+/-0,27	115,38+/-0,71	115,44+/-0,68	115,19+/-1,11	114,85+/-0,96	114,25+/-1,8	114,66+/-1,09	114,04+/-1,52	114,19+/-1,34	1,40%
L_100	Grau de Similaridade (%)	59,1+/-1,47	59,4+/-1,52	59,57+/-1,69	59,83+/-1,75	59,34+/-1,62	59,2+/-2,01	59,2+/-2,33	59,47+/-2,11	59,57+/-2,39	59,2+/-2,1	59,27+/-2,38	0,29%
	Score	229,58+/-0,52	229,47+/-0,59	229,56+/-0,46	229,34+/-0,63	229,09+/-0,74	228,83+/-0,8	228,46+/-1,31	228,52+/-1,06	227,97+/-1,1	228,26+/-1,24	228,81+/-2,37	1,21%
L_150	Grau de Similaridade (%)	56,49+/-2,01	55,91+/-1,84	56,82+/-2,05	55,87+/-1,96	56,62+/-1,45	55,98+/-2,27	56,82+/-2,41	55,78+/-1,61	56,62+/-2,16	56,84+/-2,35	56,4+/-1,95	0,16%
	Score	320,96+/-0,74	320,42+/-0,71	320,74+/-0,53	320,53+/-1,21	320,3+/-1,04	319,54+/-1,09	319,59+/-1,31	319,52+/-1,5	319,11+/-1,8	318,85+/-1,72	318,49+/-2,3	0,77%
L_200	Grau de Similaridade (%)	53,27+/-1,98	53,85+/-2,08	53,53+/-1,44	53,77+/-1,9	54,33+/-1,81	53,17+/-1,55	53,5+/-1,81	53,6+/-2,11	53,6+/-2,24	54,42+/-2,13	54,13+/-2,12	1,61%
	Score	458,29+/-1,19	458,21+/-1,64	458,31+/-1,25	458,43+/-1,36	457,36+/-1,29	457,54+/-1,79	457,3+/-1,67	457,32+/-1,97	456,34+/-3,4	456,44+/-2,51	455,27+/-2,61	0,66%

INSTÂNCIAS	ATRIBUTOS	20 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	54,8+/-3,17	53,93+/-3,52	53,6+/-3,59	53,8+/-3,74	53+/-3,57	54+/-3,35	54,2+/-2,98	54+/-3,93	54,87+/-3,75	56,6+/-3,9	54,27+/-5,77	0,97%
	Score	115,79+/-0,42	115,81+/-0,38	115,75+/-0,4	115,36+/-0,8	115,16+/-0,98	114,99+/-0,79	114,65+/-1,11	113,91+/-1,19	112,97+/-1,84	112,72+/-2,06	112,01+/-2,7	3,28%
L_100	Grau de Similaridade (%)	59,33+/-1,8	59,17+/-1,88	59,7+/-1,92	59,2+/-2,21	60,07+/-2	59,43+/-2,45	59,5+/-2,01	58,87+/-2,01	59,3+/-2,45	59,3+/-2,51	59,07+/-1,63	0,44%
	Score	229,48+/-0,59	229,52+/-0,65	229,23+/-0,59	228,93+/-0,88	228,52+/-1,04	228,63+/-0,84	228,11+/-0,87	227,71+/-1,24	227,27+/-1,63	227,43+/-1,99	226,12+/-2,29	1,46%
L_150	Grau de Similaridade (%)	56,87+/-1,44	56,71+/-1,82	56,56+/-2,06	56,38+/-2,09	56,8+/-2,38	56,78+/-2,07	56,6+/-2,19	57,07+/-2,73	57,29+/-2,22	57+/-2,28	57,27+/-1,86	0,70%
	Score	320,5+/-0,97	320,81+/-0,77	320,49+/-0,83	320,13+/-1,06	320,03+/-1,07	319,87+/-1,35	319,63+/-1,22	318,83+/-2,07	318,5+/-1,75	317,29+/-2,12	316,62+/-3,32	1,21%
L_200	Grau de Similaridade (%)	53,6+/-1,94	53,2+/-1,91	53,62+/-1,87	53,28+/-2,01	54,03+/-1,59	53,98+/-2	54,05+/-2,2	54,62+/-2,16	54+/-2,11	54,17+/-1,66	54,27+/-2,45	1,25%
	Score	458,49+/-1,12	458,74+/-1,22	458,59+/-1,11	457,95+/-1,16	457,54+/-1,46	456,79+/-2,18	456,5+/-1,82	455,63+/-1,74	454,93+/-2,52	455,35+/-2,43	455,39+/-2,88	0,68%

INSTÂNCIAS	ATRIBUTOS	30 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	54,4+/-3,63	54,33+/-3,62	54,47+/-3,29	55,87+/-3,34	55,87+/-3,69	56,93+/-3,68	57,27+/-3,44	58,33+/-4,93	59,6+/-5,92	57,13+/-5,86	60,2+/-5,85	10,66%
	Score	115,69+/-0,64	115,73+/-0,51	115,55+/-0,64	115,34+/-0,77	114,61+/-1,29	114,4+/-1,28	113,39+/-1,99	112,52+/-2,23	111,3+/-2,86	111,24+/-2,14	109,67+/-2,64	5,20%
L_100	Grau de Similaridade (%)	58,9+/-2,18	59,17+/-1,92	59,93+/-1,82	60,07+/-2,57	59,73+/-2,42	59,8+/-2,89	60,03+/-2,71	60,43+/-2,55	60,87+/-3,06	60,47+/-2,6	60,63+/-2,87	2,94%
	Score	229,41+/-0,69	229,32+/-0,67	229,33+/-0,67	228,81+/-0,79	228,21+/-1,12	227,1+/-1,56	226,99+/-1,51	226,24+/-2,17	224,51+/-2,94	224,65+/-2,77	223,48+/-2,91	2,58%
L_150	Grau de Similaridade (%)	56,58+/-2,18	56,4+/-1,95	57,31+/-2,13	56,96+/-2,14	57,71+/-2,16	58,07+/-1,86	56,87+/-2,1	57,91+/-2,59	58,71+/-2,59	59,49+/-3,04	59,93+/-3,1	5,92%
	Score	320,52+/-0,81	320,44+/-0,69	320,32+/-1	319,94+/-0,97	319,68+/-1,29	318,63+/-1,45	318,14+/-1,48	316,58+/-2,48	315,67+/-2,59	314,2+/-3,28	312,05+/-4,06	2,64%
L_200	Grau de Similaridade (%)	53,9+/-1,94	53,2+/-2	54,28+/-1,75	53,95+/-2,24	54,57+/-1,83	54,6+/-2,25	54,18+/-1,68	55,07+/-2,26	55,87+/-2,52	55,75+/-2,15	55,65+/-2,66	3,25%
	Score	458,69+/-1,47	458,3+/-1,29	458,04+/-1,33	457,69+/-1,32	457,07+/-1,52	456,69+/-2,08	455,69+/-2,23	455,32+/-2,82	453,21+/-3,23	451,56+/-2,89	451,42+/-5,12	1,58%

INSTÂNCIAS	ATRIBUTOS	40 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	53,4+/-3,47	54,8+/-3,95	56,27+/-3,3	58,53+/-2,87	57,87+/-4,76	61,2+/-4,75	62,93+/-4,89	65,53+/-5,65	65,67+/-6,61	67,27+/-7,13	68,87+/-6,98	28,97%
	Score	115,8+/-0,36	115,4+/-0,87	115,44+/-0,73	115,06+/-0,98	114,08+/-1,48	112,81+/-1,61	111,14+/-1,83	110,16+/-2,54	108,67+/-3,02	107,28+/-3	106,48+/-3,98	8,05%
L_100	Grau de Similaridade (%)	59,43+/-2,04	60,1+/-2,01	60,77+/-2,12	60,37+/-2,55	61,33+/-3,45	61,77+/-3,23	62,6+/-2,78	62,57+/-2,73	63,27+/-3,6	63,97+/-3,6	63,67+/-4,25	7,13%
	Score	229,39+/-0,57	229,49+/-0,49	228,79+/-0,87	228,2+/-1	227,33+/-1,45	226,44+/-2,01	225,2+/-2,62	224,45+/-2,07	222,78+/-3,5	221,28+/-3,36	221,12+/-3,23	3,61%
L_150	Grau de Similaridade (%)	56,13+/-1,92	56,31+/-1,48	57,69+/-1,64	57,42+/-2,27	57,78+/-2,52	58,42+/-2,09	59,53+/-3,09	60,07+/-2,57	60,58+/-2,79	60,89+/-2,77	62,04+/-3,27	10,53%
	Score	320,64+/-0,89	320,35+/-0,75	320,5+/-1,06	320,11+/-0,83	319,04+/-1,37	317,87+/-2,13	316,7+/-1,83	315,61+/-2,72	313,76+/-2,31	311,7+/-4,14	308,58+/-4,4	3,76%
L_200	Grau de Similaridade (%)	53,4+/-1,6	53,93+/-2,03	54,5+/-1,67	54,88+/-2,24	54,8+/-2,36	55,5+/-1,72	55,75+/-2,26	57,02+/-2,75	57,13+/-2,68	57,25+/-2,22	56,85+/-2,94	6,48%
	Score	458,51+/-1,04	458,6+/-1,15	457,95+/-0,99	457,24+/-1,56	457+/-1,91	454,65+/-2,17	454,11+/-1,85	452,2+/-2,95	451,76+/-3,2	448,68+/-4,78	447,31+/-5,8	2,44%

INSTÂNCIAS	ATRIBUTOS	50 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	53,13+/-3,13	57,4+/-2,14	59,13+/-2,46	63,27+/-2,61	69,67+/-3,65	72,93+/-2,57	76,2+/-2,55	81,13+/-2,95	83,53+/-2,4	84,6+/-1,56	84,93+/-1,24	59,85%
	Score	115,71+/-0,5	115,86+/-0,42	115,46+/-0,94	114,24+/-1,13	111,27+/-1,93	109,54+/-1,53	107,47+/-1,62	103,91+/-2,27	101,63+/-2,03	100,8+/-1,57	100,52+/-1,35	13,13%
L_100	Grau de Similaridade (%)	58,4+/-1,78	59,93+/-1,71	61,6+/-2,18	61,33+/-2,43	62,23+/-2,97	62,97+/-2,81	63,8+/-3,4	63,73+/-3,17	65,57+/-4,28	66,1+/-3,92	65,6+/-4,62	12,33%
	Score	229,66+/-0,49	229,51+/-0,53	228,87+/-0,88	228,42+/-1,13	227,38+/-1,53	226,04+/-1,36	224,58+/-2,96	223,61+/-2,78	221,1+/-3,41	219,25+/-4,19	217,67+/-3,87	5,22%
L_150	Grau de Similaridade (%)	56,38+/-1,95	56,71+/-2,42	58,24+/-2,21	58,73+/-2,31	59,29+/-2,32	60,82+/-2,08	61,36+/-2,05	61,87+/-3,16	61,91+/-3,32	62,4+/-3,55	63,87+/-3,11	13,28%
	Score	320,58+/-0,97	320,54+/-0,88	319,91+/-0,8	319,24+/-1,18	318,51+/-1,45	315,94+/-2,1	314,64+/-2,94	313,05+/-2,88	311,63+/-3,01	310,63+/-3,89	307,2+/-4,34	4,17%
L_200	Grau de Similaridade (%)	53,53+/-1,7	53,8+/-1,72	54,65+/-1,98	56+/-1,27	55,87+/-2,26	56,9+/-2,36	57,55+/-2,14	57,9+/-2,13	57,92+/-3,36	59,32+/-2,44	58,57+/-3,02	9,42%
	Score	458,53+/-1,09	458,37+/-1,21	457,87+/-1,21	456,98+/-1,76	455,84+/-2,01	453,56+/-2,07	451,79+/-3,11	450,27+/-2,81	449,03+/-3,94	446,21+/-4,39	444,25+/-4,64	3,11%

INSTÂNCIAS	ATRIBUTOS	60 ITERAÇÕES										FS PP	
		( $\alpha = 1 / \beta = 0$ )	( $\alpha = 1 / \beta = 0,1$ )	( $\alpha = 1 / \beta = 0,2$ )	( $\alpha = 1 / \beta = 0,3$ )	( $\alpha = 1 / \beta = 0,4$ )	( $\alpha = 1 / \beta = 0,5$ )	( $\alpha = 1 / \beta = 0,6$ )	( $\alpha = 0 / \beta = 0,7$ )	( $\alpha = 1 / \beta = 0,8$ )	( $\alpha = 1 / \beta = 0,9$ )		( $\alpha = 1 / \beta = 1$ )
L_50	Grau de Similaridade (%)	54,27+/-3,91	57,6+/-2,65	59,73+/-2,52	63,73+/-3,13	68,67+/-2,61	72,53+/-4,1	76,33+/-3,14	80,53+/-2,96	82,8+/-2,71	83,73+/-3,3	85,33+/-1,74	57,23%
	Score	115,87+/-0,21	115,69+/-0,62	115,21+/-1,1	113,77+/-1,44	111,8+/-1,59	109,57+/-2,06	107,3+/-1,7	104,41+/-2,1	102,52+/-2,2	101,29+/-1,74	99,89+/-1,5	13,79%
L_100	Grau de Similaridade (%)	59,47+/-1,69	61,07+/-1,81	62,23+/-2,23	63,27+/-2,72	64,33+/-2,78	65,77+/-3,57	67,4+/-3,65	66,4+/-3,9	70,17+/-3,62	69,3+/-3,83	71,43+/-4,4	20,11%
	Score	229,62+/-0,44	229,42+/-0,56	228,73+/-0,9	227,49+/-1,43	225,97+/-1,5	224,42+/-1,86	222,04+/-2,82	220,98+/-3,42	217,87+/-3,38	217,66+/-3,45	214,47+/-4,9	6,60%
L_150	Grau de Similaridade (%)	55,91+/-1,63	57,89+/-1,96	58,6+/-1,79	60,07+/-3,06	60,71+/-2,67	60,82+/-2,15	61,51+/-2,84	63,6+/-3,75	64+/-3,12	65,71+/-3,49	66,02+/-3,63	18,08%
	Score	320,42+/-0,75	320,64+/-1	3									

70 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	53,4+/-3,76	57,73+/-2,05	59,67+/-2,92	63,2+/-2,51	68,6+/-3,47	73,2+/-2,71	77,2+/-2,81	81,33+/-2,84	83,8+/-1,89	84,53+/-1,54	85,27+/-0,96	59,68%
	Score	115,83+/-0,32	115,83+/-0,36	115,28+/-1,06	114,22+/-1,2	112,06+/-1,68	109,55+/-1,66	106,94+/-1,88	103,97+/-2,22	101,85+/-1,63	101,21+/-1,41	100,56+/-1,1	13,18%
L_100	Grau de Similaridade (%)	59,57+/-1,69	61,63+/-2,41	62,93+/-2,45	64,8+/-2,89	66,63+/-2,92	68,2+/-2,76	68,87+/-3,96	69,53+/-4,06	71,9+/-4,56	72,9+/-4,87	73,07+/-4,81	22,66%
	Score	229,64+/-0,52	229,31+/-0,7	228,47+/-1,03	227,25+/-1,53	225,44+/-1,82	222,97+/-2,36	222+/-2,63	220,33+/-3,62	217,36+/-3,76	214,76+/-4,45	213,4+/-4,45	7,07%
L_150	Grau de Similaridade (%)	56,04+/-1,99	58,56+/-2,22	59,67+/-1,58	60,78+/-2,06	61,13+/-2,87	63,36+/-2,88	65+/-3,35	65,38+/-3,19	65,78+/-4,06	67,62+/-3,12	67,18+/-3,76	19,88%
	Score	320,77+/-0,75	320,57+/-1,01	319,55+/-1,27	318,54+/-1,55	316,58+/-1,94	314,32+/-1,95	312,01+/-3,92	310,05+/-3,15	307,1+/-3,57	304,87+/-3,24	301,92+/-4,07	5,88%
L_200	Grau de Similaridade (%)	52,8+/-1,81	54,42+/-1,95	55,48+/-2	56,68+/-2,52	58,22+/-1,94	58,95+/-1,91	59,68+/-2,09	60,7+/-2,94	61,1+/-2,95	62,15+/-3,33	63,18+/-4,08	19,66%
	Score	458,64+/-1,04	457,93+/-1,01	457,51+/-1,25	456,1+/-1,98	454,43+/-2,02	452,2+/-2,45	449,65+/-3,28	447,28+/-4,42	444,7+/-4,46	441,11+/-5,07	437,7+/-6,02	4,57%
80 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	54,27+/-3,04	57,87+/-1,26	59,4+/-2,54	63,33+/-2,7	69,8+/-3,63	72,53+/-2,68	77,8+/-1,89	81,47+/-2,87	83,27+/-1,9	84+/-1,86	85,47+/-0,88	57,49%
	Score	115,83+/-0,47	115,89+/-0,28	115,45+/-0,97	114,17+/-1,24	111,41+/-1,87	109,94+/-1,59	106,6+/-1,42	103,71+/-2,3	102,38+/-1,66	101,65+/-1,68	100,36+/-0,86	13,36%
L_100	Grau de Similaridade (%)	59,47+/-1,71	61,7+/-2,13	63,83+/-2,27	65,87+/-2,94	68,03+/-3,53	70,6+/-2,81	72,33+/-3,74	73,6+/-3,91	75,73+/-3,28	77,07+/-4,02	76,77+/-3,84	29,09%
	Score	229,39+/-0,68	229,36+/-0,73	228,49+/-1	226,4+/-1,6	224,85+/-2,3	222,23+/-2,35	219,94+/-3,2	217,19+/-3,76	214,42+/-3	210,9+/-5,25	211,19+/-3,89	7,93%
L_150	Grau de Similaridade (%)	56,22+/-2,06	58,53+/-1,76	60,24+/-2,22	62,11+/-2,22	63,33+/-2,36	64,84+/-3,25	66,04+/-3,75	67,49+/-3,72	68,42+/-3,05	71+/-3,18	69,96+/-3,89	24,44%
	Score	320,63+/-0,96	320,38+/-0,69	319,45+/-1,16	318+/-1,55	315,97+/-1,82	313,77+/-2,52	310,16+/-3,38	307,94+/-3,51	305,29+/-3,3	302,44+/-4,19	300,75+/-4,93	6,20%
L_200	Grau de Similaridade (%)	53,15+/-1,46	55,17+/-1,76	55,8+/-1,94	56,95+/-1,77	59,20+/-2,07	60,05+/-2,65	61,42+/-3,1	61,9+/-3,07	62,17+/-3	62,63+/-3,45	64,73+/-2,97	21,79%
	Score	458,31+/-1,23	457,99+/-1	456,94+/-1,56	455,84+/-2,25	454,1+/-2,11	450,32+/-3,18	447,72+/-4,02	445,39+/-4,28	441,38+/-3,47	439,99+/-5,5	434,4+/-5,87	5,22%
90 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	53,93+/-3,52	58,2+/-1,81	59,27+/-2,85	65,13+/-3	68,8+/-3,04	73,4+/-2,64	76,87+/-2,29	82,47+/-2,46	83,87+/-1,36	84,47+/-1,61	85,67+/-0,75	58,85%
	Score	115,81+/-0,34	115,75+/-0,57	115,26+/-1,12	113,28+/-1,43	111,92+/-1,46	109,34+/-1,71	107,33+/-1,58	102,99+/-1,92	101,78+/-1,29	101,22+/-1,53	100,17+/-0,71	13,50%
L_100	Grau de Similaridade (%)	58,7+/-1,55	62,13+/-1,5	64,8+/-2,81	68,9+/-2,17	70,13+/-2,38	71,3+/-3,35	74,53+/-2,95	76,93+/-3,15	77,83+/-2,77	79,17+/-2,57	80,27+/-3,38	36,75%
	Score	229,39+/-0,51	229,47+/-0,6	228,13+/-1,21	226,62+/-1,5	224,22+/-1,75	222,28+/-2,8	218,44+/-2,76	215,29+/-2,74	212,87+/-3,24	210,84+/-2,91	208,42+/-4,18	9,14%
L_150	Grau de Similaridade (%)	55,56+/-2,08	58,62+/-2,13	60,31+/-2,04	62,64+/-2,68	64,56+/-2,48	66,78+/-2,79	67,78+/-3,41	69,58+/-2,88	72,27+/-3,01	73,44+/-4,03	73,07+/-3,3	31,52%
	Score	320,53+/-0,72	320,25+/-0,89	319,5+/-1,23	317,57+/-1,7	314,66+/-1,84	312,23+/-2,66	310,44+/-3,17	305,93+/-3,54	302,7+/-2,93	299,05+/-4,73	297,97+/-5,29	7,93%
L_200	Grau de Similaridade (%)	53,27+/-2,1	55,45+/-1,68	56,22+/-2,18	57,68+/-2,07	60+/-2	61,32+/-2,11	62,65+/-3,06	63,37+/-2,47	65,05+/-2,32	65,2+/-3,43	65,25+/-3,2	22,49%
	Score	458,01+/-0,89	457,51+/-1,3	457,28+/-1,66	455,47+/-2,11	452,53+/-2,18	449,42+/-3,2	446+/-3,84	442,56+/-4,24	438,5+/-4,03	435,99+/-6,35	434,78+/-6,38	5,07%
100 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	53,13+/-3,41	57,33+/-1,49	60,47+/-3,37	64,27+/-2,62	69,6+/-3,07	72,8+/-2,45	77+/-2,86	81,6+/-2,85	83,13+/-2,29	84,47+/-1,52	85,67+/-0,96	60,49%
	Score	115,6+/-0,64	115,81+/-0,38	114,94+/-1,36	113,81+/-1,28	111,6+/-1,52	109,77+/-1,51	107,12+/-2,03	103,69+/-2,27	102,45+/-1,93	101,26+/-1,39	100,52+/-0,99	13,04%
L_100	Grau de Similaridade (%)	59+/-1,51	63,6+/-1,6	67,17+/-1,92	70,1+/-2,18	73,13+/-1,73	75,93+/-2,39	79,77+/-1,26	81,03+/-0,87	81,37+/-1,71	82,03+/-0,66	82,4+/-0,71	39,66%
	Score	229,51+/-0,55	229,13+/-0,73	227,59+/-0,98	225,46+/-1,73	222,95+/-1,76	220,11+/-2,45	215,49+/-1,67	213,71+/-1,75	212,94+/-1,86	211,57+/-2,57	210,9+/-1,87	8,11%
L_150	Grau de Similaridade (%)	56,73+/-1,81	58,89+/-1,74	61,22+/-2,67	63,51+/-2,16	66,84+/-2,7	67,11+/-3,22	70,49+/-3,54	72,69+/-3,22	74,13+/-3,63	74,47+/-3,03	75,49+/-3,19	33,07%
	Score	320,47+/-0,8	320,56+/-0,73	318,77+/-1,54	317,56+/-1,6	314,23+/-2,13	311,51+/-2,25	307,87+/-3,48	304,33+/-3,06	299,38+/-5,35	299,05+/-4,05	295,64+/-4,86	7,75%
L_200	Grau de Similaridade (%)	53,82+/-1,55	55,03+/-1,9	56,53+/-2,11	58,97+/-2,34	59,72+/-2,31	62,57+/-2,31	62,95+/-2,3	64,27+/-2,52	66,27+/-3,52	66,37+/-3,48	68,23+/-3,48	26,83%
	Score	458,18+/-1,41	458,67+/-1,79	456,66+/-1,49	454,41+/-2,06	452,82+/-2,89	449,03+/-3,44	446,2+/-3,96	442,56+/-5,23	434,7+/-5,14	434,02+/-6,78	428,52+/-7,13	6,47%
200 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	54,13+/-2,87	57,73+/-1,34	59,33+/-2,27	64,33+/-3,14	70,27+/-3,17	72,13+/-2,78	77,6+/-1,67	80,87+/-2,95	83,93+/-1,5	84,53+/-1,63	85,47+/-0,88	57,90%
	Score	115,75+/-0,49	115,75+/-0,63	115,41+/-0,9	113,83+/-1,38	111,11+/-1,71	110,13+/-1,7	106,76+/-1,24	104,23+/-2,26	101,79+/-1,4	101,16+/-1,48	100,38+/-0,92	13,28%
L_100	Grau de Similaridade (%)	59,17+/-1,69	63,9+/-1,74	66,63+/-2,07	69,47+/-1,98	72,47+/-1,63	76,8+/-1,92	79,73+/-1,44	80,93+/-0,36	81,37+/-0,55	81,83+/-0,64	82,3+/-0,53	39,09%
	Score	229,54+/-0,65	229,08+/-0,78	227,89+/-1,24	226,1+/-1,57	223,64+/-1,54	219,3+/-2,25	215,99+/-1,86	214,48+/-0,66	213,6+/-1,1	212,69+/-1,5	211,7+/-1,22	7,74%
L_150	Grau de Similaridade (%)	56,11+/-2,09	61,78+/-1,87	66,96+/-2,2	71,13+/-2,42	75,27+/-1,85	79,22+/-1,95	82,62+/-1,07	83,49+/-1	84,53+/-0,78	85,27+/-0,93	86+/-0,77	53,27%
	Score	320,82+/-0,81	319,85+/-1,17	317,49+/-1,76	314,03+/-2,2	309,34+/-2,24	303,75+/-3,39	298,04+/-1,96	296,1+/-2,25	293,35+/-2,27	291,17+/-2,91	288,31+/-2,57	10,13%
L_200	Grau de Similaridade (%)	53,77+/-1,58	58,68+/-2,6	62,3+/-1,91	66,18+/-2,45	68,97+/-4,21	70,23+/-5,69	72,98+/-5,3	73,85+/-5,23	77,07+/-4,51	74,48+/-5,95	74,75+/-7,48	39,02%
	Score	458,53+/-1,14	452,99+/-6,24	448,25+/-10,92	437,23+/-15,77	428,78+/-14,63	416,05+/-13,66	416,66+/-11,33	404,92+/-15,06	402,85+/-11,6	395,18+/-11,19	391,01+/-13,16	14,73%
300 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
L_50	Grau de Similaridade (%)	52,53+/-3,5	57,27+/-2,16	59,93+/-3,29	63,8+/-2,89	69,47+/-3,3	72,93+/-2,72	77,53+/-1,84	80,6+/-3,06	83,2+/-1,9	84,8+/-1,6	85,47+/-0,88	62,71%
	Score	115,43+/-0,78	115,71+/-0,59	115,24+/-1,28	113,9+/-1,31	111,67+/-1,59	109,66+/-1,72	106,87+/-1,37	104,42+/-2,34	102,36+/-1,66	100,96+/-1,47	100,38+/-0,89	13,04%
L_100	Grau de Similaridade (%)	59+/-1,84	63,47+/-2,16	66,87+/-1,89	69,8+/-1,78	72,87+/-1,8	76,4+/-1,52	79,9+/-1,3	81+/-0,52	81,4+/-0,55	81,73+/-0,68	82,33+/-0,6	39,54%
	Score	229,5+/-0,59	229,18+/-0,89	227,78+/-1,29	225,91+/-1,45	223,54+/-1,54	219,92+/-1,68	215,82+/-1,57	214,31+/-1	213,54+/-1,18	212,92+/-1,57	211,7+/-1,33	7,76%
L_150	Grau de Similaridade (%)	56,44+/-2	61,76+/-1,95	66,58+/-1,4	71,27+/-1,83	75,62+/-1,94	79,47+/-1,76	82,64+/-0,95	83,6+/-1,07	84,56+/-0,6	85,24+/-0,66	86,27+/-0,87	52,85%
	Score	320,77+/-0,75	320,23+/-1,05	317,56+/-1,41	313,98+/-1,92	309,26+/-2,37	303,77+/-3,02	298,46+/-1,89	296,2+/-2,62	293,98+/-1,88	291,91+/-1,81	288,67+/-2,8	10,11%
L_200	Grau de Similaridade (%)	54,03+/-2,57	58,42+/-1,96	63,12+/-2,04	67,7+/-1,69	72,17+/-1,51	75,68+/-1,9	79,17+/-1,43	81,55+/-1,23	83,2+/-1,05	84,28+/-0,88	85,22+/-0,77	57,73%
	Score	458,3											

600 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
I_50	Grau de Similaridade (%)	53,6+/-3,03	57,8+/-2,15	59,2+/-2,23	64,2+/-2,55	70,8+/-2,71	72,67+/-2,75	77,6+/-2,22	81,27+/-2,85	83,2+/-2,66	84,27+/-1,34	85,53+/-0,85	59,57%
	Score	115,85+/-0,37	115,63+/-0,65	115,33+/-0,9	113,73+/-1,24	110,94+/-1,42	109,9+/-1,66	106,78+/-1,53	104+/-2,21	102,26+/-2,17	101,49+/-1,23	100,33+/-0,83	13,40%
I_100	Grau de Similaridade (%)	59,07+/-1,21	63,5+/-2,13	66,9+/-2,21	69,4+/-1,69	73,3+/-1,86	76,9+/-2,01	79,9+/-1,14	80,57+/-0,76	81,3+/-0,53	81,83+/-0,64	82,37+/-0,66	39,44%
	Score	229,46+/-0,54	229,2+/-0,82	227,79+/-1,48	226,26+/-1,32	223+/-1,73	219,25+/-2,09	215,8+/-1,39	214,97+/-1,07	213,72+/-1,11	212,71+/-1,4	211,55+/-1,51	7,81%
I_150	Grau de Similaridade (%)	56,02+/-1,35	61,62+/-1,67	66,11+/-1,81	70,98+/-1,55	75,64+/-1,65	79,58+/-2,03	82,44+/-1,09	83,4+/-0,93	84,53+/-0,98	85,22+/-0,88	86,22+/-0,8	53,91%
	Score	320,41+/-0,99	320,12+/-0,86	317,95+/-1,41	314,66+/-1,81	309,41+/-2,07	303,72+/-3,03	298,76+/-2,23	296,88+/-2,34	293,96+/-2,69	292,28+/-2,6	289,05+/-3,09	9,79%
I_200	Grau de Similaridade (%)	53,13+/-1,47	58,43+/-2,03	63,72+/-1,79	67,58+/-2,06	72,03+/-1,69	75,32+/-1,33	79,22+/-2,03	81,48+/-1,67	83,43+/-0,97	84,58+/-0,83	85,3+/-0,67	60,55%
	Score	458,42+/-1,42	458,02+/-1,48	454,04+/-2,12	449,95+/-3,06	443,44+/-3,32	437,1+/-2,78	428,15+/-5,55	421,76+/-5,26	415,61+/-3,85	411,46+/-3,34	408,78+/-4,43	10,83%

700 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
I_50	Grau de Similaridade (%)	54,8+/-3,33	58,4+/-2,15	60,67+/-3,74	63,4+/-2,69	69,87+/-3,38	72,93+/-2,91	77,53+/-2,23	81,33+/-2,75	83,73+/-1,91	84,33+/-1,8	85,67+/-0,75	56,33%
	Score	115,89+/-0,25	115,51+/-0,75	114,88+/-1,52	114,2+/-1,27	111,42+/-1,58	109,62+/-1,79	106,79+/-1,6	103,86+/-2,15	101,96+/-1,66	101,42+/-1,61	100,2+/-0,73	13,54%
I_100	Grau de Similaridade (%)	58,9+/-1,6	63,2+/-1,87	66,8+/-2,1	70,1+/-1,97	73,3+/-2,28	76,5+/-1,63	80,2+/-1,05	80,83+/-0,58	81+/-0,63	82,03+/-0,71	82,2+/-0,54	39,56%
	Score	229,26+/-0,69	229,21+/-0,76	227,89+/-1,32	225,69+/-1,73	223,01+/-2,1	219,65+/-1,74	215,41+/-1,43	214,57+/-1,06	214,22+/-1,25	212,2+/-1,65	211,98+/-1,22	7,54%
I_150	Grau de Similaridade (%)	55,89+/-1,85	60,93+/-1,31	66,87+/-2,09	71,73+/-1,97	75,78+/-1,7	79,51+/-1,55	82,2+/-1,02	83,91+/-0,95	84,76+/-0,87	85,22+/-0,9	86,33+/-0,82	54,46%
	Score	320,78+/-0,68	320,15+/-0,83	317,85+/-1,59	313,5+/-2,14	308,93+/-2,08	303,64+/-2,36	299,25+/-1,97	295,74+/-2,67	293,76+/-2,28	292,26+/-2,79	288,86+/-2,62	9,95%
I_200	Grau de Similaridade (%)	53,78+/-1,95	58,05+/-1,83	63,28+/-1,97	67,73+/-2,01	71,55+/-1,89	76+/-1,72	79,15+/-1,52	81,92+/-1,36	83,57+/-1,24	84,45+/-0,93	85,23+/-0,8	58,48%
	Score	458,31+/-1,07	458,14+/-1,49	453,89+/-1,92	449,67+/-2,94	444,13+/-3	435,63+/-4,27	427,88+/-4,51	420,66+/-4,4	415,29+/-4,33	411,74+/-4,26	408,92+/-4,32	10,78%

800 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
I_50	Grau de Similaridade (%)	53,2+/-3,49	58,2+/-1,49	58,87+/-2,35	64+/-2,31	69,93+/-2,76	73,4+/-2,08	77,47+/-1,54	81,27+/-2,45	84,33+/-1,47	84,53+/-1,93	85,47+/-0,88	60,66%
	Score	115,83+/-0,29	115,75+/-0,57	115,52+/-0,94	113,89+/-1,13	111,31+/-1,43	109,45+/-1,39	106,95+/-1,11	104+/-1,84	101,36+/-1,34	101,19+/-1,68	100,38+/-0,86	13,34%
I_100	Grau de Similaridade (%)	58,67+/-2,21	63,47+/-1,96	66,23+/-1,89	69,37+/-1,28	73,03+/-1,99	77+/-1,9	80,37+/-1,3	80,67+/-0,7	81,3+/-0,59	81,7+/-0,64	82,37+/-0,48	40,40%
	Score	229,52+/-0,59	229,09+/-0,73	228,19+/-1,1	226,4+/-1,04	223,26+/-1,8	219,22+/-2,09	215,13+/-1,9	214,86+/-1,01	213,74+/-1,18	213,07+/-1,36	211,64+/-1,1	7,79%
I_150	Grau de Similaridade (%)	55,96+/-1,68	61,51+/-1,71	66,18+/-1,92	71,24+/-1,62	76,02+/-1,85	79,84+/-1,75	82,62+/-1,34	83,71+/-1,14	84,31+/-0,89	85,6+/-0,8	86,24+/-0,8	54,11%
	Score	320,82+/-0,87	320,08+/-0,98	318,07+/-1,36	314,12+/-2,08	308,77+/-2,53	303,37+/-2,68	298,42+/-2,56	296,24+/-2,51	294,98+/-2,53	291,38+/-2,15	289,11+/-2,82	9,88%
I_200	Grau de Similaridade (%)	53,62+/-1,9	59,18+/-1,88	62,98+/-1,73	68,07+/-2,03	72,94+/-1,75	75,75+/-2,09	79,33+/-1,32	81,65+/-1,27	83,97+/-0,91	84,77+/-0,62	85,45+/-0,64	59,36%
	Score	458,67+/-1,27	457,02+/-1,61	454,63+/-2,36	449,26+/-2,89	441,54+/-3,65	436,13+/-4,45	427,85+/-3,87	421,32+/-4,28	413,93+/-4,08	411,23+/-3,49	408,05+/-3,64	11,04%

900 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
I_50	Grau de Similaridade (%)	53,67+/-2,83	57,67+/-1,64	59,53+/-3	63,47+/-2,58	69,07+/-3,82	73,47+/-2,96	77,47+/-2,31	80,8+/-2,95	83,6+/-1,58	84,13+/-1,54	85,67+/-0,75	59,03%
	Score	115,86+/-0,26	115,78+/-0,48	115,26+/-1,06	114,07+/-1,12	111,76+/-1,77	109,39+/-1,88	106,88+/-1,63	104,22+/-2,33	102,11+/-1,44	101,6+/-1,37	100,18+/-0,74	13,53%
I_100	Grau de Similaridade (%)	59,07+/-1,63	63,7+/-1,93	66,37+/-1,92	69,83+/-1,24	73,07+/-1,95	76,27+/-2,06	80,23+/-1,12	80,97+/-0,6	81,23+/-0,5	81,77+/-0,5	82,2+/-0,7	39,16%
	Score	229,52+/-0,47	229,14+/-0,76	228,06+/-1,17	226,03+/-1,12	223,21+/-1,91	219,88+/-2,33	215,28+/-1,43	214,16+/-1,08	213,89+/-1,1	212,89+/-1,05	211,94+/-1,65	7,66%
I_150	Grau de Similaridade (%)	56,16+/-1,59	61,64+/-1,68	65,93+/-2,06	70,44+/-2,03	75,87+/-2,04	79,84+/-1,94	82,58+/-1,3	83,49+/-0,91	84,67+/-0,93	85,16+/-0,79	86,04+/-0,71	53,21%
	Score	320,62+/-0,79	320,13+/-0,74	318,04+/-1,64	314,89+/-2,03	308,69+/-2,7	303,33+/-2,99	298,54+/-2,47	296,89+/-2,12	294,01+/-2,51	292,66+/-2,34	289,86+/-2,66	9,59%
I_200	Grau de Similaridade (%)	53,78+/-1,83	58,02+/-1,98	63,02+/-1,69	67,95+/-1,82	71,77+/-1,56	76,17+/-1,74	79,1+/-1,83	81,58+/-1,63	83,88+/-0,83	84,62+/-0,75	85,23+/-0,73	58,48%
	Score	458,59+/-1,07	456,82+/-1,73	455,41+/-2,02	449,15+/-2,51	443,54+/-3,14	435,07+/-4,16	428+/-4,79	421,87+/-4,91	414,57+/-4,07	412,31+/-3,87	409,48+/-4,27	10,71%

1000 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	FS PP
I_50	Grau de Similaridade (%)	53,6+/-2,65	57,87+/-1,86	60+/-3,1	63,47+/-2,58	69,27+/-3,08	72,93+/-3,34	76,47+/-2,51	81,93+/-2,22	83,2+/-1,51	85,13+/-1,43	85,6+/-0,8	59,70%
	Score	115,84+/-0,38	115,75+/-0,57	115,16+/-1,19	114,14+/-1,18	111,74+/-1,69	109,67+/-1,94	107,54+/-1,83	103,45+/-1,66	102,42+/-1,36	100,68+/-1,3	100,22+/-0,8	13,48%
I_100	Grau de Similaridade (%)	59,2+/-2,04	63,57+/-1,65	67,07+/-2,21	69,77+/-1,61	72,93+/-2,11	76,5+/-1,63	80,17+/-1,27	80,83+/-0,45	81,2+/-0,48	81,7+/-0,64	82,33+/-0,6	39,07%
	Score	229,4+/-0,62	229,25+/-0,8	227,54+/-1,26	226,07+/-1,26	223,14+/-1,98	219,87+/-1,73	215,43+/-1,7	214,67+/-0,82	213,93+/-0,97	213,04+/-1,42	211,68+/-1,38	7,72%
I_150	Grau de Similaridade (%)	55,62+/-1,92	61,42+/-1,98	66,04+/-1,63	70,73+/-1,9	76+/-1,31	79,31+/-1,24	82,38+/-1,45	83,73+/-0,78	84,62+/-0,97	85,38+/-0,67	86,29+/-0,86	55,14%
	Score	320,7+/-0,73	320,24+/-0,98	318,39+/-1,25	314,67+/-2,01	308,97+/-1,86	304,18+/-1,63	299,04+/-2,83	296,19+/-2,11	294,01+/-2,53	291,94+/-2	289,18+/-2,76	9,83%
I_200	Grau de Similaridade (%)	53,3+/-1,63	58,17+/-1,94	63,62+/-1,93	68,68+/-2,02	72,52+/-1,57	75,73+/-1,72	79,02+/-1,35	81,7+/-1,33	83,37+/-0,87	84,7+/-0,94	85,22+/-0,89	59,89%
	Score	458,53+/-1,38	457,72+/-1,35	454,84+/-2,04	448,17+/-3,46	442,61+/-3,26	436,67+/-3,89	428,49+/-3,82	421,43+/-3,96	416,53+/-3,65	411,32+/-4,15	409,2+/-4,92	10,76%

## APÊNDICE B – Resultados utilizando MLP

		10 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	53,87+/-3,01	53,13+/-3,37	54,13+/-3,22	54,93+/-3,38	54,6+/-3,39	55,67+/-3,87	55,73+/-3,68	55,6+/-4,08	55,8+/-4,27	55,53+/-5	57,53+/-4,02	6,79%
	Score	115,88+/-0,21	115,69+/-0,57	115,57+/-0,59	115,32+/-0,99	114,75+/-1,25	114,32+/-1,4	113,95+/-1,53	113,06+/-1,64	112,96+/-1,52	112,74+/-1,7	110,96+/-2,58	4,25%
L_100	Grau de Similaridade (%)	58,9+/-1,27	59,87+/-2	59,43+/-1,8	59,87+/-2,05	60,23+/-2,06	59,87+/-2,64	59,5+/-2,31	59,83+/-2,37	60,4+/-3,1	60,73+/-2,68	59,63+/-2,59	1,24%
	Score	229,51+/-0,62	229,36+/-0,49	229,39+/-0,55	228,78+/-1,02	228,19+/-1,04	227,85+/-1,31	227,11+/-1,82	226,54+/-1,68	226,02+/-2,15	225,04+/-2,34	224,1+/-2,55	2,36%
L_150	Grau de Similaridade (%)	56,22+/-1,82	56,53+/-1,82	56,84+/-1,68	57,11+/-1,82	56,53+/-2,32	57,33+/-2,2	57,78+/-2,39	57,93+/-2,44	57,29+/-2,2	57,71+/-1,95	58,47+/-2,55	4,00%
	Score	320,54+/-0,78	320,73+/-0,73	320,47+/-0,79	320,16+/-1,16	319,76+/-1,24	319,07+/-1,47	318,2+/-1,59	317,19+/-2,04	317,22+/-2,18	315,69+/-2,63	314,18+/-2,88	1,98%
L_200	Grau de Similaridade (%)	53,1+/-1,97	53,9+/-1,96	53,6+/-2,09	54,55+/-1,69	54,52+/-2,03	55,12+/-2,15	54,85+/-2,18	54,65+/-2,06	55,37+/-2,19	54,98+/-2,38	54,5+/-2,04	2,84%
	Score	458,5+/-1,24	458,46+/-1,44	457,98+/-1,47	457,54+/-1,53	456,7+/-1,57	456,7+/-2,09	455,18+/-2,85	455,12+/-2,62	454,63+/-3,22	453,86+/-2,64	452,63+/-3,59	1,28%

		20 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	54,07+/-3,71	54,07+/-3,03	55,47+/-3,69	57,33+/-3,24	57,33+/-4,17	59,87+/-3,86	60,6+/-4,17	60,33+/-4,29	62,4+/-4,24	63,6+/-5,17	64,93+/-4,28	20,09%
	Score	115,72+/-0,52	115,74+/-0,37	115,32+/-0,9	114,86+/-1,13	114,16+/-1,23	112,88+/-1,25	112,62+/-1,78	112,08+/-1,91	110,57+/-1,86	110,29+/-2,45	109,05+/-2,55	5,76%
L_100	Grau de Similaridade (%)	59,2+/-1,8	59,77+/-1,87	59,73+/-1,82	60,93+/-1,59	61,47+/-2,31	61,83+/-2,32	62,33+/-2,26	62,23+/-2,7	62,1+/-2,41	62,87+/-2,75	63,37+/-3,15	7,04%
	Score	229,26+/-0,6	229,52+/-0,53	228,75+/-0,91	228,62+/-1,2	227,34+/-1,44	226,84+/-1,58	225,91+/-1,95	225,03+/-2,13	224,12+/-2,62	223,51+/-2,91	221,84+/-3,01	3,24%
L_150	Grau de Similaridade (%)	55,6+/-1,93	56,98+/-1,64	57,18+/-1,79	57,13+/-2,22	59,04+/-1,93	58,44+/-2,3	58,82+/-2,45	59,56+/-2,29	60,13+/-2,5	60,53+/-2,76	60,78+/-2,24	8,92%
	Score	320,7+/-0,87	320,47+/-0,84	320,42+/-0,81	319,74+/-1,13	318,6+/-1,45	318,01+/-1,56	316,42+/-2,27	314,62+/-2,03	315,04+/-3,02	312,91+/-2,78	312,13+/-2,71	2,67%
L_200	Grau de Similaridade (%)	53,17+/-1,79	54,02+/-1,25	54,2+/-1,95	54,97+/-2,25	55,47+/-2,22	55,62+/-1,82	57+/-2,04	56+/-2,45	56,48+/-2,21	56,4+/-1,45	56,17+/-2,7	5,84%
	Score	458,55+/-1,21	458,83+/-1,21	457,96+/-1,28	456,7+/-1,45	455,78+/-1,88	455,3+/-1,77	453,56+/-2,16	453,22+/-2,43	451,39+/-2,96	451,6+/-2,57	449,53+/-3,59	1,97%

		30 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	54+/-3,76	56,4+/-2,44	57,8+/-2,27	59,07+/-3,38	60,07+/-3,2	62,27+/-3,41	64,47+/-5,05	65,07+/-3,53	68,07+/-3,98	68+/-4,03	66,87+/-3,96	23,83%
	Score	115,82+/-0,47	115,94+/-0,21	115,37+/-0,74	114,68+/-1,22	114,13+/-1,25	112,91+/-1,43	111,29+/-1,84	111,16+/-1,58	109,71+/-1,69	109,27+/-1,95	108,67+/-2,07	6,17%
L_100	Grau de Similaridade (%)	58,63+/-1,76	60,33+/-1,53	60,8+/-2,15	61,73+/-2,17	62,5+/-2,08	63,67+/-2,68	64,37+/-2,75	63,97+/-3,16	64,63+/-3,1	65,9+/-3,47	66,73+/-3,47	12,40%
	Score	229,37+/-0,63	229,17+/-0,69	228,94+/-0,8	227,96+/-1,19	227,2+/-1,27	226,21+/-1,44	225,03+/-1,77	223,34+/-1,96	222,68+/-2	221,7+/-2,34	220,95+/-2,43	3,67%
L_150	Grau de Similaridade (%)	56,4+/-1,72	57,49+/-1,84	57,87+/-1,87	58,76+/-2,21	59,42+/-2,63	59,89+/-1,87	61,09+/-2,72	61,18+/-2,43	61,38+/-2,21	62,47+/-2,33	62,87+/-3,05	11,47%
	Score	320,52+/-0,84	320,68+/-0,72	319,87+/-1,08	319,11+/-1,28	318,58+/-1,3	316,79+/-1,78	315,32+/-2,3	313,8+/-1,73	312,73+/-2,31	311,22+/-3,37	309,14+/-3,04	3,55%
L_200	Grau de Similaridade (%)	53,28+/-1,81	53,96+/-1,82	54,68+/-2,44	55,6+/-2,27	55,93+/-2,2	56,12+/-1,96	56,88+/-1,77	57,48+/-2,04	57,77+/-2,21	58,82+/-1,99	58,53+/-2,07	9,85%
	Score	458,43+/-1,27	458,04+/-1,36	458,68+/-2,44	456,58+/-1,83	455,51+/-2,05	454,29+/-2,59	452,83+/-2,35	451,24+/-3,52	449,72+/-2,67	446,52+/-3,28	446,71+/-3,9	2,56%

		40 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	54,33+/-3,43	57,6+/-2,15	58,47+/-1,98	61,07+/-2,86	63,4+/-3,51	66,93+/-2,77	68,47+/-3,08	69,2+/-3,45	72+/-3,1	72,27+/-3,38	74,27+/-4,06	36,70%
	Score	115,82+/-0,39	115,85+/-0,34	115,42+/-0,71	114,34+/-1,1	113,3+/-1,38	111,88+/-1,39	110,99+/-1,67	109,91+/-1,69	108,71+/-1,56	108,57+/-2,06	106,71+/-2,49	7,87%
L_100	Grau de Similaridade (%)	59,23+/-1,75	60,6+/-2,08	62+/-2,1	62,93+/-2,13	63,7+/-2,37	64,03+/-3,06	65,43+/-2,89	66,5+/-2,9	67,53+/-2,94	68,97+/-2,54	67,53+/-2,94	14,01%
	Score	229,61+/-0,55	229,43+/-0,59	228,71+/-0,89	227,72+/-1,23	226,96+/-1,58	226,21+/-1,69	224,53+/-2,15	223,15+/-2,09	221,78+/-2,35	221,17+/-2,65	219,74+/-2,79	4,30%
L_150	Grau de Similaridade (%)	55,96+/-1,85	57,49+/-1,52	58,22+/-2,13	59,82+/-1,98	60,58+/-2,45	61,22+/-2,32	62+/-3,02	62,18+/-2,79	62,93+/-2,5	64,58+/-2,77	64,2+/-2,55	14,72%
	Score	320,68+/-0,85	320,68+/-0,77	320,04+/-0,98	319,41+/-1,32	317,75+/-1,51	315,91+/-1,98	313,92+/-2,8	312,92+/-1,87	310,99+/-2,79	309,66+/-2,82	308,04+/-2,4	3,94%
L_200	Grau de Similaridade (%)	53,12+/-1,74	53,95+/-2	55,62+/-1,88	56,32+/-2	57,08+/-1,48	58,45+/-1,9	58,6+/-1,99	59,35+/-2,49	59,08+/-2,35	59,4+/-3,05	60,05+/-2,12	13,05%
	Score	458,28+/-1,21	457,92+/-1,25	457,32+/-1,68	456,43+/-1,42	454,74+/-1,82	452,26+/-2,28	451,76+/-2,41	449+/-3,08	447,47+/-3,01	446,26+/-4,78	443,47+/-3,89	3,23%

		50 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	53,6+/-3,03	58,2+/-1,08	59,33+/-2,44	62,6+/-2,69	65,4+/-3,02	68+/-2,83	69,27+/-2,5	72,8+/-2,9	72,73+/-2,94	74,73+/-2,85	76+/-3,06	41,79%
	Score	115,83+/-0,39	115,79+/-0,57	115,41+/-0,92	114,27+/-1,18	113,2+/-1,4	111,72+/-1,52	111,17+/-1,5	109,48+/-1,76	109,26+/-1,59	107,96+/-1,87	107,11+/-1,79	7,53%
L_100	Grau de Similaridade (%)	59,3+/-1,79	61,83+/-1,83	62,3+/-2,05	64,17+/-2,1	64,9+/-1,85	65,93+/-2,38	66,03+/-2,98	67,67+/-2,9	68,1+/-2,74	68,73+/-2,69	70,5+/-1,96	18,89%
	Score	229,36+/-0,65	229,04+/-0,7	228,68+/-0,94	227,68+/-1,34	226,51+/-1,72	225,3+/-1,81	224,07+/-2,05	221,98+/-2,65	221,74+/-1,93	220,91+/-2,47	219,05+/-2,44	4,50%
L_150	Grau de Similaridade (%)	56,09+/-1,3	57,84+/-1,98	59,29+/-1,95	60,67+/-1,7	61,47+/-2,25	62,76+/-2,28	63,62+/-2,36	63,84+/-3,26	64,6+/-2,22	65,73+/-2,81	66,42+/-2,44	18,42%
	Score	320,73+/-0,72	320,79+/-0,9	319,45+/-0,94	318,22+/-1,3	317,15+/-1,59	315,52+/-1,51	314,69+/-2,15	313,02+/-2,69	311,42+/-2,7	309,47+/-4,08	307,12+/-2,43	4,24%
L_200	Grau de Similaridade (%)	53,55+/-1,93	55+/-1,58	56,32+/-1,94	57,03+/-1,66	57,82+/-2,07	58,55+/-1,83	58,9+/-2,06	60,93+/-2,27	60,12+/-2,17	61,57+/-2,14	61,33+/-2,21	14,53%
	Score	458,36+/-1,3	458,18+/-1,35	456,65+/-1,42	456,95+/-1,81	454,4+/-2,27	452,54+/-2,45	451,12+/-2,86	447,77+/-3,08	447,86+/-3,88	445,23+/-3,56	442,04+/-4,79	3,56%

		60 ITERAÇÕES											Fator de Similaridade Preço da Preferência
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	
L_50	Grau de Similaridade (%)	53,4+/-2,79	57,73+/-1,34	59+/-1,84	62,93+/-2,72	65,6+/-2,94	68,93+/-2,62	70,47+/-2,91	73,27+/-2,39	74,4+/-2,98	75,93+/-2,8	77,2+/-2,86	44,57%
	Score	115,79+/-0,4	115,82+/-0,53	115,55+/-0,62	114,17+/-1,22	113,16+/-1,37	111,71+/-1,36	110,92+/-1,59	109,31+/-1,53	108,61+/-1,94	107,35+/-2,13	106,72+/-1,93	7,83%
L_100	Grau de Similaridade (%)	58,93+/-1,88	62,03+/-2,06	63,57+/-2,29	64,53+/-2,78	65,97+/-2,26	66,93+/-2,86	67,77+/-2,89	69,53+/-2,54	70,7+/-3,12	71,33+/-2,16	71,93+/-2,53	22,06%
	Score	229,36+/-0,65	229,23+/-0,82	228,64+/-0,97	227,28+/-1,25	226,15+/-1,6	225,12+/-1,74	224,03+/-1,98	222,12+/-1,99	220,74+/-1,99	219,42+/-2,39	218,89+/-2,86	4,58%
L_150	Grau de Similaridade (%)	56,44+/-1,11	58,16+/-1,54	59,91+/-2,31	61,69+/-2,05	62,49+/-2,53	63,38+/-2,35	64,24+/-2,66	66,62+/-2,41	66,09+/-1,84	67,02+/-3,32	68,2+/-2,71	20,84%
	Score	320,79+/-0,											

70 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	53,93+/-2,9	58,07+/-1,09	59,24+/-2,23	63,27+/-2,28	65,53+/-2,35	69,42+/-2,97	70,87+/-2,91	73,13+/-1,91	74,8+/-2,61	77,13+/-2,62	77,53+/-2,11	43,76%
	Score	115,78+/-0,38	115,74+/-0,5	115,39+/-0,94	114,01+/-1,04	113,21+/-1,13	111,58+/-1,48	110,76+/-1,59	109,43+/-1,2	106,39+/-1,8	106,84+/-1,96	106,49+/-1,49	8,02%
I_100	Grau de Similaridade (%)	58,63+/-1,87	61,47+/-1,86	63,67+/-1,78	65,33+/-1,74	67,1+/-2,18	67,37+/-2,17	69,47+/-2,55	70,93+/-2,32	71,77+/-2,6	72,3+/-2,77	74+/-2,84	26,22%
	Score	229,49+/-0,53	229,28+/-0,79	228,45+/-1,02	227,59+/-1,19	226,17+/-1,43	225,48+/-1,11	223,5+/-1,98	222,08+/-2,07	220,83+/-2,09	220,42+/-2,25	218,13+/-2,96	4,95%
I_150	Grau de Similaridade (%)	55,78+/-1,41	58,31+/-1,49	60,69+/-1,85	61,96+/-2,4	64,47+/-1,81	65,11+/-2,29	65,73+/-2,15	67,09+/-2,62	67,82+/-2,01	67,4+/-2,66	69,29+/-2,22	24,22%
	Score	320,7+/-0,46	320,57+/-0,82	319,71+/-0,88	318,1+/-1,56	315,86+/-1,73	314,32+/-1,69	313,37+/-2,46	310,7+/-2,5	309,78+/-3,28	308,46+/-2,72	306,08+/-3,44	4,56%
I_200	Grau de Similaridade (%)	53,05+/-1,34	55,15+/-2,4	56,63+/-1,86	58,82+/-1,7	59,13+/-2,09	59,83+/-2,12	61,5+/-2,32	61,52+/-2,15	61,9+/-2,82	62,7+/-2,3	63,78+/-2,09	20,23%
	Score	458,34+/-1,29	458,21+/-1,21	456,8+/-1,42	454,94+/-2,11	453,56+/-2,01	451,92+/-2,88	449,25+/-2,86	447,04+/-2,89	445,5+/-3,93	444,84+/-3,74	441,52+/-4,01	3,67%

80 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	53,67+/-2,83	57,8+/-1,81	59,24+/-2,86	62,87+/-2,35	66,6+/-2,59	70,13+/-2,53	71,33+/-2,49	74+/-2,78	74,8+/-2,71	76,13+/-2,68	76,2+/-3,16	45,71%
	Score	115,85+/-0,32	115,71+/-0,5	115,41+/-0,99	114,3+/-1,03	112,76+/-1,16	111,19+/-1,34	110,54+/-1,37	108,86+/-1,7	108,5+/-1,8	107,43+/-1,84	106,06+/-2,44	8,45%
I_100	Grau de Similaridade (%)	59,43+/-1,54	62,57+/-1,75	64,5+/-1,93	65,63+/-2,14	68,1+/-2,01	69,97+/-1,91	71,13+/-2,26	72,83+/-2,24	73,03+/-1,66	74,4+/-2,52	74,77+/-2,32	25,81%
	Score	229,62+/-0,54	229,32+/-0,46	228,16+/-1,21	227,72+/-1,25	225,63+/-1,28	224,1+/-1,75	223,24+/-2	221,29+/-2,18	220,94+/-1,9	219,22+/-2,6	218,97+/-2,31	4,63%
I_150	Grau de Similaridade (%)	56,07+/-1,86	58,13+/-1,43	61,16+/-1,95	63,27+/-2,19	63,78+/-2,45	66,13+/-2,15	67,24+/-2,69	67,82+/-2,21	68,87+/-3,07	69,96+/-3,08	70,73+/-2,81	26,51%
	Score	320,8+/-0,88	320,63+/-1,02	319,46+/-1,12	317,52+/-1,41	315,99+/-1,6	314,09+/-1,67	312,44+/-2,57	310,26+/-2,25	308,45+/-3,15	307,4+/-2,69	305,14+/-3,39	4,51%
I_200	Grau de Similaridade (%)	53,53+/-1,91	55,4+/-1,71	57,03+/-1,62	58,67+/-2,07	59,28+/-2,1	60,13+/-2,24	61,55+/-2,1	62,35+/-1,5	63,32+/-2,49	63,65+/-2,01	63,82+/-3,23	19,22%
	Score	458,69+/-1,12	458,42+/-1,39	457,1+/-1,7	455,05+/-2,45	453,54+/-2,32	451,34+/-2,89	448,72+/-3,01	446,88+/-2,72	445,93+/-3,29	443,86+/-4,49	440,86+/-5,45	3,89%

90 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	53,2+/-3,41	57,87+/-1,63	60,13+/-2,96	63+/-2,41	66,6+/-2,74	69,47+/-2,92	71,07+/-2,91	73,73+/-2,35	75,13+/-2,86	77,73+/-2,35	79,27+/-2,71	49,00%
	Score	115,61+/-0,61	115,77+/-0,51	115,11+/-1,25	114,23+/-1,15	112,83+/-1,35	111,57+/-1,44	110,55+/-1,59	108,98+/-1,54	108,22+/-1,92	106,36+/-1,66	105,2+/-2,01	9,00%
I_100	Grau de Similaridade (%)	58,83+/-1,97	63+/-1,69	64,9+/-2,3	66,57+/-1,86	68,33+/-1,83	70,4+/-1,96	71,33+/-2,29	73,3+/-1,81	75,03+/-2,37	75,17+/-2,5	76,1+/-2,61	29,36%
	Score	229,62+/-0,54	229,26+/-0,7	228,34+/-1,18	227,55+/-1,21	226,02+/-1,18	224,64+/-1,8	223,54+/-2,04	221,7+/-1,83	219,76+/-2,88	219,14+/-2,17	217,92+/-2,59	5,10%
I_150	Grau de Similaridade (%)	55,91+/-1,61	58,58+/-1,47	61,4+/-2,12	63,27+/-2,15	65,36+/-2,27	66,53+/-1,86	67,69+/-2,7	69,38+/-1,82	70,67+/-2,53	71,62+/-2,04	72,02+/-2,58	28,81%
	Score	320,7+/-0,88	320,63+/-1,02	319,46+/-1,12	317,52+/-1,41	315,99+/-1,6	314,09+/-1,67	312,44+/-2,57	310,26+/-2,25	308,45+/-3,15	307,4+/-2,69	305,14+/-3,39	4,88%
I_200	Grau de Similaridade (%)	52,9+/-1,68	55,78+/-1,79	57,57+/-2,25	59,12+/-1,82	60,08+/-1,76	61,82+/-1,25	63,18+/-1,82	63,27+/-1,82	63,88+/-2,24	64,35+/-2,2	64,48+/-2,27	21,89%
	Score	458,24+/-1,19	457,76+/-1,36	456,76+/-1,44	455,06+/-1,67	453,01+/-1,88	451,35+/-1,95	447,97+/-3,54	446,62+/-2,77	445,3+/-3,22	443,43+/-3,76	441,94+/-4,97	3,57%

100 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	53,13+/-3,33	58,13+/-1,63	59,53+/-2,62	63+/-2,41	65,93+/-1,75	71,27+/-1,97	71,6+/-2,03	74,8+/-2,66	75,13+/-3,04	77,33+/-2,21	79+/-2,05	49,69%
	Score	115,76+/-0,35	115,78+/-0,38	115,31+/-1,1	114,4+/-1,05	113,13+/-1,1	110,58+/-1,05	110,29+/-1,1	108,46+/-1,75	108,16+/-2,08	106,88+/-1,57	105,38+/-1,5	8,97%
I_100	Grau de Similaridade (%)	59+/-1,51	63,6+/-1,6	67,17+/-1,92	70,1+/-2,18	73,13+/-1,73	75,93+/-2,39	79,77+/-1,26	81,03+/-0,87	81,37+/-0,71	82,03+/-0,66	82,4+/-0,71	39,66%
	Score	229,32+/-0,55	229,13+/-0,73	227,59+/-0,98	225,46+/-1,73	222,95+/-1,76	220,11+/-2,45	215,49+/-1,67	213,71+/-1,75	212,94+/-1,86	211,57+/-1,77	210,94+/-1,87	8,11%
I_150	Grau de Similaridade (%)	56,73+/-1,81	58,89+/-1,74	61,22+/-2,67	63,51+/-2,16	66,84+/-2,17	67,11+/-3,22	70,49+/-3,54	72,69+/-3,22	74,13+/-3,63	74,47+/-3,03	75,49+/-3,19	33,07%
	Score	320,47+/-0,8	320,56+/-0,73	318,77+/-1,54	317,56+/-1,6	314,23+/-2,13	311,51+/-3,25	307,87+/-3,48	304,33+/-3,06	299,38+/-5,35	299,05+/-4,05	295,64+/-4,86	7,75%
I_200	Grau de Similaridade (%)	53,82+/-1,55	55,03+/-1,9	56,53+/-2,11	58,97+/-2,34	59,72+/-2,31	62,57+/-2,41	62,95+/-2,3	64,27+/-2,56	67,27+/-2,52	66,37+/-3,52	69,23+/-3,48	28,63%
	Score	458,18+/-1,41	458,67+/-1,79	456,66+/-1,49	454,41+/-2,06	452,82+/-2,89	449,03+/-3,44	446,2+/-3,96	442,56+/-5,23	434,7+/-5,14	434,02+/-6,78	428,52+/-7,13	6,47%

200 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	54,33+/-3,35	58,33+/-1,37	59,33+/-2,6	62,73+/-1,59	66,87+/-2,57	70,73+/-2,34	72,67+/-2,27	76,47+/-2,62	77,67+/-2,07	79,4+/-2,64	80,8+/-2,45	48,72%
	Score	115,79+/-0,4	115,61+/-0,72	115,38+/-1,03	114,47+/-0,93	112,81+/-1,3	110,93+/-1,23	109,79+/-1,33	107,35+/-1,82	106,71+/-1,48	105,29+/-1,93	104,09+/-2,02	10,10%
I_100	Grau de Similaridade (%)	58,37+/-1,68	63,4+/-1,6	65,9+/-1,9	68,47+/-2	70,97+/-1,82	71,8+/-1,42	73,73+/-2,13	75,37+/-1,28	78+/-1,55	79,33+/-1,7	80,57+/-1,36	36,03%
	Score	229,32+/-0,67	229,39+/-0,66	228,33+/-1,16	226,66+/-1,38	224,87+/-1,47	224,16+/-1,41	222,38+/-2,08	220,81+/-1,43	217,7+/-2,08	216,33+/-2,23	214,54+/-1,86	6,83%
I_150	Grau de Similaridade (%)	56,07+/-2,01	60,78+/-1,67	64,22+/-1,3	67,02+/-1,77	69,27+/-1,91	71,48+/-1,6	73,27+/-1,72	74,89+/-1,79	76,67+/-1,78	77,38+/-2,01	78,38+/-1,84	39,79%
	Score	320,72+/-0,77	320,35+/-0,94	319,31+/-1,3	317,15+/-1,44	315,36+/-1,75	313,42+/-1,93	311,32+/-2,09	309,29+/-2,34	306,74+/-2,4	305,23+/-2,92	304,43+/-2,67	5,20%
I_200	Grau de Similaridade (%)	53,37+/-1,57	57,38+/-2,02	60,47+/-1,83	63,35+/-2,01	65,02+/-1,54	66,58+/-1,84	67,77+/-1,82	68,97+/-1,89	69,48+/-1,79	70,73+/-2,26	72,07+/-1,91	35,04%
	Score	458,38+/-1,43	457,66+/-1,57	455,64+/-2,07	453,36+/-2,47	451,42+/-2,03	449,09+/-2,89	447,42+/-3,32	444,84+/-2,95	444,03+/-3,46	441,74+/-4,74	438,73+/-3,67	4,29%

300 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
I_50	Grau de Similaridade (%)	53,2+/-3,33	58+/-1,46	59,6+/-2,65	63,13+/-2,77	68,93+/-3	70,93+/-1,98	73,07+/-1,84	76,07+/-2,76	77,73+/-2,62	79,93+/-2,8	81,6+/-2,5	53,38%
	Score	115,79+/-0,4	115,61+/-0,72	115,38+/-1,03	114,12+/-1,23	111,76+/-1,36	110,81+/-1,23	109,62+/-1,21	107,72+/-1,83	106,61+/-1,89	104,88+/-2,13	103,49+/-1,94	10,51%
I_100	Grau de Similaridade (%)	58,4+/-1,85	63,33+/-1,42	66,7+/-1,72	68,07+/-1,21	70,23+/-2,03	72,37+/-2,09	74,67+/-1,58	76,57+/-1,96	77,93+/-1,53	79,93+/-1,59	80,43+/-1,5	37,72%
	Score	229,31+/-0,65	229,31+/-0,63	227,82+/-0,9	227,13+/-0,67	225,41+/-1,85	223,45+/-2	221,33+/-1,78	219,4+/-2,27	217,94+/-1,73	215,22+/-2,1	214,7+/-2,18	6,37%
I_150	Grau de Similaridade (%)	56,22+/-1,56	60,62+/-1,58	64,27+/-1,94	66,91+/-1,57	69,87+/-1,52	71,67+/-1,89	73,49+/-1,75	75,36+/-1,77	76,11+/-1,65	77,87+/-2	78,82+/-1,49	40,20%
	Score	320,56+/-0,79	320,04+/-0,97	319,09+/-1,36	317,49+/-1,47	315,1+/-1,75	313,01+/-2,36	310,72+/-2,23	308				

600 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
L <sub>50</sub>	Grau de Similaridade (%)	52,84±3,82	57,84±1,49	58,87±2,11	63,07±2,72	69±3,86	71,84±1,74	74,73±2,94	77,93±2,61	80,07±2,9	81,93±2,22	84,27±1,34	59,60%
	Score	115,68±0,62	115,77±0,46	115,51±0,82	114,22±1,25	111,77±1,76	110,36±0,97	108,57±1,93	106,3±1,77	104,84±2,38	103,4±1,77	101,35±1,32	12,39%
L <sub>100</sub>	Grau de Similaridade (%)	59,64±1,28	63,97±2,17	66,53±1,77	68,03±1,64	70,44±1,98	72,53±2,06	75±1,88	76,4±1,91	78,53±1,43	80,73±1,29	81,07±1,03	36,02%
	Score	229,45±0,69	228,83±0,94	228,06±1,09	227,27±1,02	225,38±1,17	223,63±1,83	221,3±1,93	219,65±2,12	217,31±1,74	214,37±2,01	213,62±1,31	6,81%
L <sub>150</sub>	Grau de Similaridade (%)	55,69±1,77	60,53±1,67	64,16±1,74	66,96±1,84	69,07±1,79	71,44±1,6	73±1,76	73,96±1,56	75,78±1,75	76,8±1,73	77,29±2,22	38,79%
	Score	320,56±0,93	320,48±0,55	319,2±1,38	317,49±1,46	315,72±1,79	313,38±1,9	311,04±2,56	309,83±2,17	307,48±2,4	305,88±2,46	304,82±3,53	4,91%
L <sub>200</sub>	Grau de Similaridade (%)	52,73±1,46	57,73±1,63	61,23±1,83	63,85±1,41	65,93±1,84	67,48±1,74	68,45±1,62	69,62±1,67	71,02±1,81	71,53±1,51	73,13±1,77	38,69%
	Score	458,45±1,52	457,74±1,38	456,01±1,46	453,72±1,61	451,23±1,93	449,87±2,22	447,15±3,14	445,23±2,86	443,01±3,68	442,36±2,61	437,71±3,21	4,52%

700 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
L <sub>50</sub>	Grau de Similaridade (%)	53,47±3,58	57,87±2,12	59,2±2,56	63,2±2,56	69,6±3,81	72,73±2,61	75,73±3,09	77,67±2,13	79,47±2,36	82,73±2,1	83,73±1,98	56,59%
	Score	115,78±0,37	115,63±0,61	115,49±1,05	114,16±1,26	111,42±1,69	109,77±1,55	107,79±2,07	106,61±1,65	105,31±1,89	102,71±1,84	101,88±1,7	12,01%
L <sub>100</sub>	Grau de Similaridade (%)	59,43±1,87	63,37±1,74	66,4±2,3	68,93±1,15	70,6±1,69	73,3±1,83	75,43±2,5	77,4±1,56	78,87±1,78	80,2±1,35	81,37±0,8	36,92%
	Score	229,38±0,64	229,23±0,72	228±1,15	226,76±0,86	225,27±1,57	222,86±1,69	220,66±2,47	218,65±1,87	216,7±2,25	215,08±2,09	213,45±1,37	6,94%
L <sub>150</sub>	Grau de Similaridade (%)	56,04±1,71	61±1,75	64,16±1,38	67,2±1,65	69,13±1,62	71,22±1,84	73,2±1,65	74,36±1,64	75,58±2,13	76,89±1,95	77,31±2,36	37,96%
	Score	320,69±0,76	320,07±0,78	319,02±1,2	317,46±1,39	315,29±1,71	313,38±2,18	310,94±2,33	309,68±2,47	307,42±2,76	306,13±3,05	305,19±3,56	4,83%
L <sub>200</sub>	Grau de Similaridade (%)	52,55±1,62	58±1,85	61,48±1,42	63,8±1,49	65,78±1,3	67,35±1,44	68,23±1,82	69,62±2,24	71,08±1,73	71,5±1,53	72,3±2,18	37,58%
	Score	458,52±1,31	457,83±1,49	455,92±1,3	453,74±1,88	451,38±2,24	449,66±2,49	447,82±2,86	444,92±3,29	442,56±3,68	441,4±3,43	439,97±4,18	4,05%

800 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
L <sub>50</sub>	Grau de Similaridade (%)	54,2±3,07	57,4±1,87	59±2,41	64,2±3,03	68,4±3,24	73±1,91	75,87±2,78	77,8±2,02	80,4±3,12	82,87±2,77	84,33±1,72	55,59%
	Score	115,72±0,55	115,83±0,45	115,45±0,98	113,73±1,3	112,03±1,55	109,52±1,18	107,86±1,82	106,6±1,55	104,52±2,36	102,6±2,32	101,26±1,48	12,50%
L <sub>100</sub>	Grau de Similaridade (%)	58,83±1,83	63,3±1,9	66±1,81	68,63±1,58	70,6±2,37	72,97±2,26	75,07±2,25	76,97±1,99	79,63±1,6	80,5±1,09	81,67±0,75	38,82%
	Score	229,49±0,56	229,1±0,73	228,32±1,09	226,6±1,05	225,06±1,93	223,14±2,18	221,09±2,33	218,97±2,33	215,81±2,11	214,71±1,48	212,69±1,29	7,32%
L <sub>150</sub>	Grau de Similaridade (%)	56,09±1,94	60,98±1,78	64,02±1,76	67,09±1,77	69,82±1,89	71,49±1,57	72,67±2,09	74,24±1,87	76,07±1,66	76,42±1,75	77,78±1,89	38,67%
	Score	320,35±0,87	320,31±0,84	319,17±1,07	317,5±1,3	314,97±1,81	313,64±1,99	311,56±2,53	309,59±2,34	307,88±2,06	306,84±2,2	305,25±2,78	4,71%
L <sub>200</sub>	Grau de Similaridade (%)	52,83±1,74	58,25±1,82	61,6±1,17	63,8±1,37	65,72±1,49	67,33±1,3	68,6±1,33	69,93±1,62	71,38±1,64	71,82±1,67	72,73±1,87	37,67%
	Score	458,26±1,15	457,8±1,22	455,41±1,86	453,92±1,95	451,85±2,29	449,15±2,15	447,2±2,08	445,07±2,35	442,38±3,02	441,06±3,07	439,74±4,34	4,05%

900 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
L <sub>50</sub>	Grau de Similaridade (%)	54,33±3,1	57,33±2,02	59,67±3,01	63,79±2,41	67,53±4,58	72,33±2,69	76,33±2,43	78,8±2,23	80,47±2,81	82,73±2,61	84,07±1,59	54,74%
	Score	115,78±0,56	115,78±0,55	115,27±1,16	114,1±1,2	112,32±2,21	110,95±1,51	107,59±1,62	105,77±1,67	104,53±2,17	102,69±2,18	101,54±1,49	12,30%
L <sub>100</sub>	Grau de Similaridade (%)	59,9±1,58	63,43±1,26	66,1±2,12	69,2±1,54	70,73±1,77	74,07±2,17	75,2±1,58	77,93±1,57	79,53±1,43	80,93±0,77	81,23±0,8	35,61%
	Score	229,57±0,53	229,19±0,84	228,11±1,53	226,31±1,3	225,05±1,48	222,22±1,95	221,06±1,72	218±1,91	216,03±1,76	214,24±1,26	213,75±1,52	6,89%
L <sub>150</sub>	Grau de Similaridade (%)	56,56±1,67	61±2,12	63,98±1,45	67,36±1,91	69,6±1,73	71,4±2,09	72,73±1,65	74,33±1,44	76,13±1,7	76,58±2,45	77,18±2,01	36,46%
	Score	320,58±0,47	320,15±1,06	319,26±1,1	317,24±1,77	315,12±1,34	313,3±2,27	312±1,96	309,85±1,75	307,58±2,22	306,62±3,03	305,26±3,33	4,78%
L <sub>200</sub>	Grau de Similaridade (%)	53,17±1,77	58,62±2,35	61,35±1,35	63,72±1,22	65,9±1,55	68,88±1,61	68,92±1,49	69,58±1,4	70,22±1,51	71,82±2,06	72,65±1,21	36,64%
	Score	458,63±1,43	457,63±1,51	456,09±1,13	453,62±1,87	451,67±2,25	449,77±2,74	446,84±2,78	445,81±1,92	444,72±2,16	441,28±4,27	439,25±2,81	4,23%

1000 INTERAÇÕES													
INSTÂNCIAS	ATRIBUTOS	( $\alpha = 1/\beta = 0$ )	( $\alpha = 1/\beta = 0,1$ )	( $\alpha = 1/\beta = 0,2$ )	( $\alpha = 1/\beta = 0,3$ )	( $\alpha = 1/\beta = 0,4$ )	( $\alpha = 1/\beta = 0,5$ )	( $\alpha = 1/\beta = 0,6$ )	( $\alpha = 0/\beta = 0,7$ )	( $\alpha = 1/\beta = 0,8$ )	( $\alpha = 1/\beta = 0,9$ )	( $\alpha = 1/\beta = 1$ )	Fator de Similaridade Preço da Preferência
L <sub>50</sub>	Grau de Similaridade (%)	52,8±3,04	57,8±1,49	59,07±2,35	63,67±2,2	68,8±3,29	72,53±2,78	77,07±1,98	79,87±2,73	80,47±2,72	83,53±2,4	84,53±1,36	60,09%
	Score	115,63±0,61	115,9±0,46	115,45±0,89	113,91±1,11	111,85±1,68	109,93±1,61	107,18±1,36	104,99±2,07	104,56±2,23	102,01±2,11	101,09±1,26	12,57%
L <sub>100</sub>	Grau de Similaridade (%)	58,7±1,53	63,1±1,64	66,3±2,31	69,07±1,44	71,4±1,93	73,63±1,91	75,83±1,59	77,7±2,02	79,8±1,45	80,9±1,01	81,57±0,76	38,96%
	Score	229,53±0,47	229,25±0,58	228,1±1,08	226,45±1,23	224,65±1,66	222,58±1,84	220,36±1,77	218,28±2,29	215,68±1,98	214,22±1,56	213,12±1,55	7,15%
L <sub>150</sub>	Grau de Similaridade (%)	56,6±1,19	61,02±1,45	64,56±1,75	67,49±1,57	69,84±1,59	71,93±1,85	73,18±1,5	75,22±1,52	75,8±2,14	77,02±1,75	78,27±1,78	38,29%
	Score	320,76±0,95	320,28±0,73	318,71±1,4	317,4±1,31	315,17±1,76	313,2±1,92	311,23±1,86	308,9±1,91	307,56±3,32	305,46±2,49	303,99±2,64	5,23%
L <sub>200</sub>	Grau de Similaridade (%)	53,48±1,83	58,17±1,66	62,02±1,43	64±1,53	66,1±1,56	67,3±1,62	68,52±1,53	70,72±1,74	70,35±1,86	71,65±1,64	72,33±1,62	35,25%
	Score	458,62±0,84	457,58±1,4	455,45±1,71	453,44±1,72	451,3±2,24	449,42±2,6	447,17±3,02	443,56±2,75	443,42±3,67	441,64±3,19	440,13±3,68	4,07%