



UNIVERSIDADE FEDERAL DA BAHIA

TESE DE DOUTORADO

**Uma Abordagem Multi-objetivo para Redução do Risco da
Insatisfação dos Stakeholders no Planejamento de Versões de
Software**

Antonio Mauricio da Silva Pitangueira

Programa de Pós-Graduação em Ciência da Computação

Salvador
03 de Novembro de 2016

ANTONIO MAURICIO DA SILVA PITANGUEIRA

**UMA ABORDAGEM MULTI-OBJETIVO PARA REDUÇÃO DO
RISCO DA INSATISFAÇÃO DOS STAKEHOLDERS NO
PLANEJAMENTO DE VERSÕES DE SOFTWARE**

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientadora: Prof^a Dra. Rita Suzana Pitangueira Maciel

Co-orientador: Prof. Dr. Márcio de Oliveira Barros

Salvador

03 de Novembro de 2016

Ficha catalográfica.

Pitangueira, A.M.S

Uma Abordagem Multi-objetivo para Redução do Risco da Insatisfação dos *Stakeholders* no Planejamento de Versões de Software / Antonio Mauricio da Silva Pitangueira– Salvador, 03 de Novembro de 2016.

135p.: il.

Orientadora: Prof^a Dra. Rita Suzana Pitangueira Maciel.

Co-orientador: Prof. Dr. Márcio de Oliveira Barros.

Tese (Doutorado)– UNIVERSIDADE FEDERAL DA BAHIA, INSTITUTO DE MATEMÁTICA, 03 de Novembro de 2016.

TOPICOS PARA FICHA CATALOGRAFICA.

I. Maciel, R.S.P. II. Oliveira, M.B..

III. UNIVERSIDADE FEDERAL DA BAHIA. INSTITUTO DE MATEMÁTICA. IV. Título.

NUMERO CDD

TERMO DE APROVAÇÃO

ANTONIO MAURICIO DA SILVA PITANGUEIRA

**UMA ABORDAGEM MULTI-OBJETIVO PARA
REDUÇÃO DO RISCO DA INSATISFAÇÃO
DOS STAKEHOLDERS NO PLANEJAMENTO
DE VERSÕES DE SOFTWARE**

Esta Tese de Doutorado foi julgada adequada à obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 03 de Novembro de 2016

Profa. Dra. Rita Suzana Pitangueira Maciel -
Orientadora
Universidade Federal da Bahia

Prof. Dr. Cláudio Nogueira Sant'Anna
Universidade Federal da Bahia

Prof. Dr. George Marconi de Araújo Lima
Universidade Federal da Bahia

Profa. Dra. Claudia Maria Lima Werner
Universidade Federal do Rio de Janeiro

Prof. Dr. Jerffeson Teixeira de Souza
Universidade Estadual do Ceará

Dedico este Trabalho a Deus, a minha família, a todos os meus professores, a meus amigos e a todos os meus alunos que fizeram parte da construção dessa jornada.

AGRADECIMENTOS

Agradeço a Deus pela presença constante em todos os momentos da minha vida, por ter me dado saúde para concluir esse caminho. E, principalmente, por me resgatar nos momentos mais difíceis. Sem Ele nada seria possível de concluir.

Aos meus orientadores Rita Maciel e Márcio Barros que me auxiliaram todo o tempo na pesquisa com ideias e sugestões, o que fez e fará de mim um melhor pesquisador.

Ao povo da minha cidade de Coração (Trento), por todo auxílio durante meu estágio doutoral. Também à FBK, principalmente, a Paolo Tonella e Angelo Susi por me indicarem o caminho de pesquisa e pela amizade. Ao meu amigo Fitsum Kifetew, por toda ajuda na parte técnica. Minha Gratidão.

Aos professores Cláudio Sant'Anna, Jerffeson Teixeira, George Marconi e Claudia Werner por participarem da minha banca de defesa de doutorado.

Aos meus pais Roque Pitangueira e Marília Pitangueira, por todo esforço que fizeram para me educar, além do amor sempre presente. Minha eterna gratidão.

À minha querida esposa Jacqueline por toda paciência, conselhos e sabedoria. Sua presença constante me faz sempre mais feliz. À minha trentininha Elena, seu sorriso filha é um banho de energia positiva.

Aos meus familiares, especialmente, minha dinda Perpétua pela amizade e conselhos que me mostram sempre uma solução inteligente.

A todos meus amigos, principalmente, Silvio, Aristóteles e Luciano por sempre apostarem na minha capacidade e fazer os meus dias mais alegres.

Aos meus companheiros do Centro Espírita Casa de Redenção Joanna de Ângelis por toda emanção de energia a meu favor. E, claro, agradeço muito aos mentores espirituais por ouvir minhas queixas, minhas dores e sempre me intuírem a ser um homem melhor.

Aos meus parceiros do Pathwork e a Márcia Almeida, por me mostrarem sempre os dois lados da vida e me conduzirem à beleza da unidade.

À CAPES, pelo apoio financeiro no meu estágio doutoral na Fondazione Bruno Kessler (FBK), Trento, Itália.

Aos meus colegas de IFBa, especialmente aos do Labrasoft, Antonio Carlos e Luiz Machado pelos conselhos sempre úteis.

A todos os meus professores e alunos, que fizeram e continuam fazendo com que eu sempre me esforce para ser um profissional cada dia melhor.

*É sábio quem vê nas dificuldades uma oportunidade de crescimento. A
força que há em ti é força de solução.*

—LOURIVAL LOPES

RESUMO

Os requisitos a serem entregues na próxima versão do software são selecionados de acordo com o valor percebido pelos *stakeholders*, o custo esperado da implementação, a disponibilidade de orçamento da companhia de software, relações de precedência entre os requisitos e restrições técnicas, entre outros. As abordagens existentes para o problema de seleção de requisitos de software não levam em conta o risco de insatisfação dos *stakeholders*, possivelmente resultante da elevada incerteza e divergência nas estimativas dos valores dos requisitos realizadas pelos *stakeholders*.

Esta Tese tem como objetivo propor uma nova abordagem multi-objetivo que considera o risco de insatisfação dos *stakeholders* no desenvolvimento da próxima versão do software (RA-MONRP). A abordagem RA-MONRP foi desenvolvida através de dois *solvers* (Yices e Z3) que adotam uma estratégia de busca determinística com base na *Satisfiability Modulo Theory* (SMT). Além disso, para permitir a aplicação da formulação proposta para problemas com número elevado de requisitos, também foi implementado um algoritmo probabilístico baseado na metaheurística genética NSGA-II.

Para avaliar a abordagem foram realizados três tipos de experimentos. O primeiro experimento verificou a existência de soluções que reduzam o risco de insatisfação dos *stakeholders* com mínimo impacto no custo e valor médio dos requisitos. O segundo experimento comparou a escalabilidade das técnicas SMT e metaheurística. Por fim, o terceiro experimento comparou o conjunto de soluções obtido por essas técnicas.

Os resultados encontrados mostram que: (i) o risco de insatisfação dos *stakeholders* pode ser minimizado com mínimo impacto no custo/valor; (ii) que a abordagem é escalável na maioria das situações em que o NSGA-II é usado, enquanto que para os *SMT solvers* a abordagem escala para problemas que não são excessivamente grandes em termos de número de requisitos e/ou não são muito esparsos em termos de número de dependências, e (iii) nos casos em que os dados apresentam alta densidade de dependências (número de dependências dividido pelo número de requisitos), o conjunto de soluções obtido pelo NSGA-II se distancia do conjunto de soluções obtido pelas técnicas baseadas em SMT. Para casos em que o número de requisitos é elevado e há baixa densidade de dependências, o NSGA-II obtém soluções muito próximas do SMT.

Os resultados obtidos indicam um avanço no estado da arte no problema de seleção de requisitos de software através da inclusão do risco de insatisfação dos *stakeholders*. Sendo assim, espera-se que ao se usar a abordagem proposta nessa Tese, os usuários possam adotar o RA-MONRP ao longo do ciclo de desenvolvimento do software, gerando um produto final de acordo com as expectativas dos *stakeholders*.

Palavras-chave: Tomada de decisão baseada no risco, Problema da Próxima Versão, Múltiplos stakeholders.

ABSTRACT

The requirements to be delivered in the next version of a software system are typically selected according to the stakeholders' perceived value, expected implementation cost, the available budget, precedence relationships between the requirements and technical constraints. Existing approaches to the requirement selection problem do not take into account the risk of stakeholders' dissatisfaction, possibly resulting from high uncertainty and divergence in the stakeholders' estimates of the value of each requirement.

This Thesis aims to propose a new multi-objective approach that considers the stakeholders' dissatisfaction risk in the development of the next software release (RA-MONRP). The approach was developed through two solvers (Yices and Z3) which adopt a deterministic search strategy based on *Satisfiability Module Theory* (SMT). In addition, to allow the application of the proposed formulation to large size problems, a probabilistic algorithm based on the NSGA-II metaheuristic was also implemented.

To assess the approach three experiments were performed. The first experiment tested for the existence of solutions that reduce the stakeholders' dissatisfaction risk with minimal impact on requirements' cost and average value. The second experiment compared the scalability of the SMT and metaheuristic techniques. Finally, the third experiment compared the set of solutions obtained by these techniques.

Experimental results show that: (i) the stakeholder dissatisfaction risk can be minimized with minimum impact on cost/value; (ii) the approach can scale to large problems in most of the cases when NSGA-II is used, while SMT solvers scale up to problems that are not too large in terms of the number of requirements and/or not too sparse in terms of dependencies; and (iii) in cases where the data set has high density (number of dependencies divided by the number of requirements), the solution set obtained by NSGA-II differs significantly from the solutions set obtained by the SMT solvers. On the other hand, NSGA-II obtains solutions very close to those found by the SMT Cases if the number of requirements is high and there is low dependency density.

The results represent an improvement over the state-of-the-art for the software requirements selection problem by taking into account the risk of stakeholders' dissatisfaction. Thus it is hoped that by using the approach proposed in this Thesis, users can adopt RA-MONRP as part of their software development life-cycle and generate final products in accordance with stakeholders' expectations.

Keywords: Risk-aware Decision Making, Next Release Problem, Multi-Stakeholder.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 O Estado da Arte	2
1.2 Problematização	3
1.3 Objetivos e Contribuições	6
1.4 Metodologia da Tese	8
1.5 Organização da Tese	10
Capítulo 2—Referencial Teórico	13
2.1 Engenharia de Requisitos	13
2.1.1 Requisitos de Software	13
2.1.2 Fases da Engenharia de Requisitos	15
2.2 Seleção de requisitos de software	18
2.2.1 Técnicas para a Seleção de Requisitos de Software Baseadas em Buscas	19
2.2.1.1 Problema da Próxima Versão	21
2.2.1.2 Técnicas baseadas em busca para o MONRP	22
2.2.2 Considerações Finais	25
Capítulo 3—Revisão e Mapeamento sistemático sobre seleção de requisitos de software usando a abordagem SBSE	27
3.1 Planejamento e Execução da Revisão Sistemática	28
3.1.1 Identificação da Pesquisa - Estratégia de Busca	30
3.1.2 Critérios e Procedimentos para a Seleção de Estudos	31
3.1.3 Avaliação da Qualidade dos estudos - Critérios adotados	33
3.2 Resultados da Revisão Sistemática e Mapeamento da Literatura	34
3.2.1 Resultados do Mapeamento da Literatura	34
3.2.2 Resultados da Revisão Sistemática	36
3.3 Resumo dos resultados da Revisão Sistemática e Mapeamento da Área	44
3.4 Considerações Finais	46
Capítulo 4—Riscos em Projetos de Software	47
4.1 Categorização dos Riscos de Software	48
4.2 Análise de Riscos de Software	49

4.3	Trabalhos Relacionados - Abordagens para Tratamento de Riscos em Seleção de Requisitos	50
4.4	Considerações Finais	53
Capítulo 5—Abordagem Proposta		55
5.1	Abordagem para o RA-MONRP com múltiplos <i>stakeholders</i>	56
5.2	Implementação das Técnicas de Busca	59
5.2.1	Implementação da Abordagem através de SMT	60
5.2.2	Implementação através da Meta-Heurística NSGA-II	62
5.3	Considerações Finais	64
Capítulo 6—Avaliação		67
6.1	Metodologia para a Experimentação da Proposta	67
6.2	Procedimento para Avaliação e Validação dos Resultados	70
6.2.1	RQ1. É possível encontrar soluções que reduzam o risco de insatisfação dos stakeholders com impacto mínimo no custo e valor dos requisitos?	71
6.2.2	RQ2. Qual a escalabilidade da abordagem quando o cenário cresce em número de requisitos, em número de dependências ou em ambos?	72
6.2.3	RQ3. Quão próximas são as soluções encontradas pela implementação da meta-heurística e da implementação baseada em SMT?	74
6.3	Resultados e Avaliação	75
6.3.1	Resultados para a Redução do Risco de Insatisfação dos <i>Stakeholders</i> - RQ1	75
6.3.1.1	Tratamento 1	75
6.3.1.2	Tratamento 2	81
6.3.1.3	Tratamento 3	85
6.3.2	Resultados para Teste de Escalabilidade das Técnicas - RQ2	87
6.3.2.1	Escalabilidade para aumento conjunto de requisitos e dependências	88
6.3.2.2	Escalabilidade para aumento apenas do número de requisitos	89
6.3.2.3	Escalabilidade para aumento apenas do número de dependências	89
6.3.3	Resultados para Comparação da Qualidade das Frentes de Pareto - RQ3	90
6.3.3.1	Resultados para diferença percentual média entre as Frentes de Pareto obtidas pelas técnicas de busca	90
6.3.3.2	Resultados para teste de aproximação das Frentes de Pareto	92
6.4	Discussão dos Resultados	94
6.4.1	Análise Consolidada sobre as Questões de Pesquisa	98
6.5	Ameaças à Validade	101
6.6	Considerações Finais	102

Capítulo 7—Conclusão e Trabalhos Futuros	103
7.1 Limitações do Trabalho	104
7.2 Trabalhos Futuros	105
Apêndice A—Estudos Selecionados para a Revisão	107
A.1 Estudos publicados em 2001	107
A.2 Estudos publicados em 2003	107
A.3 Estudos publicados em 2004	107
A.4 Estudos publicados em 2005	107
A.5 Estudos publicados em 2006	107
A.6 Estudos publicados em 2007	108
A.7 Estudos publicados em 2008	108
A.8 Estudos publicados em 2009	108
A.9 Estudos publicados em 2010	108
A.10 Estudos publicados em 2011	108
A.11 Estudos publicados em 2012	109
A.12 Estudos publicados em 2013	109
Apêndice B—Resultados Adicionais para o Tratamento 2	111
Apêndice C—Resultados Adicionais para o Tratamento 3	119

LISTA DE FIGURAS

1.1	Visão Geral do RA-MONRP	5
1.2	Metodologia da Tese	8
2.1	Três Dimensões da Engenharia de Requisitos. Adaptado de Pohl (POHL, 1993)	16
2.2	Processo incremental da Engenharia de Requisitos. Adaptado de Lamsweerde (LAMSWEERDE, 2009)	17
2.3	Arquitetura geral da Abordagem SBSE. Adaptado de Harman et al. (HARMAN et al., 2012)	20
2.4	Exemplo da Estrutura do NRP. Adaptado de Bagnall et al. (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001)	21
2.5	Exemplo de Frente de Pareto para problema de maximização de dois objetivos. Adaptado de Abraham et al.(ABRAHAM; JAIN; GOLDBERG, 2005)	23
3.1	Estágios do processo de seleção de artigos	32
3.2	Tendências das Publicações por Ano	34
3.3	Pesquisadores mais Relevantes	35
3.4	Fóruns de Publicações	35
3.5	Distribuição anual por tipo de evento	36
3.6	Distribuição por tipo de formulação do problema	37
3.7	Tipo de Abordagem para Seleção de Requisitos de Software em SBSE	39
3.8	Técnicas para Seleção de Requisitos de Software em SBSE	39
3.9	Técnicas de Busca mais utilizadas	40
3.10	Tipos de dados usados nos experimentos	41
3.11	Tipos de dados usados nos experimentos	41
3.12	Resultados para os critérios de validade geral	42
3.13	Resultados para a validade interna	43
3.14	Resultados para validade de conclusão	43
3.15	Resultados para a validade de construção	43
3.16	Resultados para validade externa	43
5.1	Visão Geral do RA-MONRP	56
5.2	Regiões de Interesse para soluções P1, P2, P3 (PITANGUEIRA et al., 2016)	57
6.1	Ilustração do cenário da abordagem para diferentes ROIs	71
6.2	Frente de Pareto (dataset-1)	76
6.3	Frente de Pareto (dataset-2)	76

6.4	Histograma de variações percentuais para a região de interesse 1 (<i>dataset-1</i>)	77
6.5	Tendência de variações para a região de interesse 1 (<i>dataset-1</i>)	78
6.6	Histograma de variações percentuais para a região de interesse 2 (<i>dataset-1</i>)	78
6.7	Tendência de variações para a região de interesse 2 (<i>dataset-1</i>)	79
6.8	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-1</i>)	79
6.9	Tendência de variações para a região de interesse 3 (<i>dataset-1</i>)	79
6.10	Histograma de variações percentuais para a região de interesse 1 (<i>dataset-2</i>)	80
6.11	Tendência de variações para a região de interesse 1 (<i>dataset-2</i>)	80
6.12	Histograma de variações percentuais para a região de interesse 2 (<i>dataset-2</i>)	81
6.13	Tendência de variações para a região de interesse 2 (<i>dataset-2</i>)	81
6.14	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-2</i>)	81
6.15	Tendência de variações para a região de interesse 3 (<i>dataset-2</i>)	82
6.16	Histograma de variações percentuais para a região de interesse 1 (<i>dataset-1</i>) com valor livre	83
6.17	Tendência de variações para a região de interesse 1 (<i>dataset-1</i>) com valor livre	84
6.18	Histograma de variações percentuais para a região de interesse 1 (<i>dataset-2</i>), com Custo Livre	84
6.19	Tendência de variações para a região de interesse 1 (<i>dataset-2</i>), com custo livre	84
6.20	Histograma de variações percentuais para a região de interesse 2 (<i>dataset-1</i>), Tratamento 3	86
6.21	Tendência de variações para a região de interesse 2 (<i>dataset-1</i>), Tratamento 3	86
6.22	Histogramas de variações de percentuais para a região de interesse 2 (<i>dataset-2</i>), Tratamento 3	87
6.23	Tendência de variações para a região de interesse 2 (<i>dataset-2</i>), Tratamento 3	87
6.24	Tendência percentual para aproximação das Frentes de Pareto (<i>dataset-2</i> escalabilidade)	94
6.25	Tendência percentual para aproximação das Frentes de Pareto (<i>dataset-2</i> escalabilidade)	95
6.26	Tendência percentual para aproximação das Frentes de Pareto (<i>dataset-2</i> escalabilidade)	96
B.1	Histograma de variações percentuais para a região de interesse 2 (<i>dataset-1</i>), com custo livre	111
B.2	Tendência de variações para a região de interesse 2 (<i>dataset-1</i>), com custo livre	112
B.3	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-1</i>), com valor livre	112
B.4	Tendência de variações para a região de interesse 3 (<i>dataset-1</i>), com valor livre	112
B.5	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-1</i>), com custo livre	113

B.6	Tendência de variações para a região de interesse 3 (<i>dataset-1</i>), com custo livre	113
B.7	Histograma de variações de percentagens para a região de interesse 1 (<i>dataset-2</i>), com valor livre	114
B.8	Tendência de variações para a região de interesse 1 (<i>dataset-2</i>), com valor livre	114
B.9	Histograma de variações de percentagens para a região de interesse 2 (<i>dataset-2</i>), com valor livre	114
B.10	Tendência de variações para a região de interesse 2 (<i>dataset-2</i>), com valor livre	114
B.11	Histograma de variações percentuais para a região de interesse 2 (<i>dataset-2</i>), com custo livre	115
B.12	Tendência de variações para a região de interesse 2 (<i>dataset-2</i>), com custo livre	115
B.13	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-2</i>), com valor livre	116
B.14	Tendência de variações para a região de interesse 3 (<i>dataset-2</i>), com valor livre	116
B.15	Histograma de variações percentuais para a região de interesse 3 (<i>dataset-2</i>), com custo livre	116
B.16	Tendência de variações para a região de interesse 3 (<i>dataset-2</i>), com custo livre	117
C.1	Histograma de variações de percentagens para o ROI 1 (<i>dataset-1</i>) - Tratamento 3	119
C.2	Tendência de variações para o ROI 1 (<i>dataset-1</i>)- Tratamento 3	119
C.3	Histogramas de variações de percentagens para o ROI 3 (<i>dataset-1</i>) - Tratamento 3	120
C.4	Tendência de variações para o ROI 3 (<i>dataset-1</i>) - Tratamento 3	120
C.5	Histogramas de variações de percentagens para o ROI 1 (<i>dataset-2</i>) - Tratamento 3	121
C.6	Tendência de variações para o ROI 1 (<i>dataset-2</i>) - Tratamento 3	121
C.7	Histogramas de variações de percentagens para o ROI 3 (<i>dataset-2</i>) - Tratamento 3	122
C.8	Tendência de variações para o ROI 3 (<i>dataset-2</i>) - Tratamento 3	122

LISTA DE TABELAS

3.1	Critérios de qualidade selecionados para avaliação dos estudos (BARROS; DIAS-NETO, 2011)	33
4.1	Abordagens para Tratamento de Riscos em Seleção de Requisitos	53
6.1	Estágios da Pesquisa desenvolvidos na Tese	67
6.2	Quadro Conceitual da Pesquisa	70
6.3	Resultados - <i>dataset-1</i>	75
6.4	Resultados - <i>dataset-2</i>	75
6.5	Resultados por região de interesse para o <i>dataset-1</i>	77
6.6	Resultados por região de interesse para o <i>dataset-2</i>	77
6.7	Dataset-1: Resultados para Tratamento 2	82
6.8	Dataset-2: Resultados para o Tratamento 2	83
6.9	Resultados por região de interesse para <i>dataset-1</i> (Tratamento 3)	85
6.10	Resultados por região de interesse para <i>dataset-2</i> (Tratamento 3)	85
6.11	Resultados para escalabilidade - aumento conjunto requisitos-dependências	88
6.12	Resultados para escalabilidade aumentando apenas o número de requisitos	89
6.13	Resultados para escalabilidade aumentando o número de dependências	90
6.14	Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II)	91
6.15	Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao crescimento do número de requisitos e dependências (REQ/DEP)	91
6.16	Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao crescimento do número de requisitos (REQ)	92
6.17	Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao aumento do número de dependências (DEP)	92
6.18	Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 para os conjuntos de dados originais	93
6.19	Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao crescimento do número de requisitos e dependências (REQ/DEP)	93

6.20	Resultados para o teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao aumento do número de requisitos (REQ)	94
6.21	Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao aumento do número de dependências (DEP)	95
6.22	Dados agrupados para tempo de execução e taxa de aproximação para os conjuntos de dados originais	99
6.23	Dados agrupados para tempo de execução e taxa de aproximação para o aumento conjunto de requisitos e dependências	100
6.24	Dados agrupados para tempo de execução e taxa de aproximação para o aumento apenas do número de requisitos	100
6.25	Dados agrupados para tempo de execução e taxa de aproximação quando do aumento apenas do número de dependências	100

Capítulo

1

INTRODUÇÃO

Requisitos de software (RS) expressam as necessidades e restrições impostas a um produto de software que contribuem para a solução de problemas no mundo real. No contexto de desenvolvimento de um software, a Engenharia de Requisitos (ER) é uma etapa essencial pois é responsável pela gestão dos requisitos durante todo o ciclo de vida do produto de software (FAIRLEY; BOURQUE, 2014; KOTONYA; SOMMERVILLE, 1998).

Durante a Engenharia de Requisitos, uma má interpretação dos requisitos ou falta de compreensão destes aumenta o risco de não satisfazer as necessidades de pessoas da organização que possuem influência direta ou indireta nos requisitos de software (SIM; BROUSE, 2014). Essas pessoas são denominadas *stakeholders* e, conforme Pohl (POHL, 2010), são as mais importantes fontes de requisitos para o desenvolvimento de um software. De um modo geral, a engenharia de requisitos é uma das etapas mais importantes para o desenvolvimento de um software porque se preocupa com a relação entre fatores estruturais, de comportamento, aspirações dos *stakeholders* e evolução do software ao longo do tempo (ZAVE, 1997; AZMEH; MIRBEL; CRESCENZO, 2013).

De acordo com Pohl (POHL, 2010), o sucesso do produto de software está fortemente relacionado com as expectativas dos *stakeholders*. Projetos de software bem sucedidos levam em consideração tanto a possibilidade dos *stakeholders* demandarem propriedades do sistema que simplifiquem e facilitem a utilização do mesmo, bem como que o software atenda às exigências e aspirações dos *stakeholders*.

Quando estes aspectos não são devidamente considerados em um projeto de software, problemas podem levar o projeto a situações de risco, como por exemplo, a falta de motivação dos *stakeholders* quando não percebem suas necessidades sendo satisfeitas, quando assumem uma postura excessivamente crítica em relação ao projeto de software e/ou quando percebem que não estão sendo contemplados em seus anseios ao longo da etapa de engenharia de requisitos (FAIRLEY; BOURQUE, 2014; LAMSWEERDE, 2009). Caso esses riscos não sejam atenuados ou resolvidos, as possíveis consequências podem ser: 1) atrasos no projeto de software, afetando a conclusão no prazo esperado para a entrega do produto; 2) produto de software em desacordo com os objetivos estabelecidos; e/ou 3)

não entrega do software (FAIRLEY; BOURQUE, 2014; FEATHER; CORNFORD, 2003; ASNAR; GIORGINI; MYLOPOULOS, 2011; MOORES; CHAMPION, 1996).

Para tratar esses riscos, a estratégia de desenvolvimento incremental tem sido utilizada para dividir o software em versões menores (ou incrementos) e, portanto, tratar os riscos em cada versão, no momento em que eles podem surgir (SOMMERVILLE, 2004; PRESSMAN, 2001). Um incremento (ou versão) é composto por um conjunto de requisitos que formam um produto operacional do software (PRESSMAN, 2001). Ao desenvolver software em incrementos, os engenheiros de requisitos e os *stakeholders* devem selecionar o conjunto de requisitos que comporá cada incremento e que será integrado na próxima versão do software (HARMAN; JONES, 2001; ZHANG; HARMAN; MANSOURI, 2007).

A seleção do conjunto de requisitos que fará parte da próxima versão do software é um processo decisório que envolve um balanceamento entre o custo de desenvolvimento e o valor da nova versão do software a ser entregue para os *stakeholders*. Esta combinação está sujeita a múltiplas restrições, muitas vezes contrastantes, tais como dependências técnicas e preferências expressas pelos *stakeholders* (DURILLO et al., 2009). Além disso, tal processo de decisão é usualmente complexo devido à presença de diferentes *stakeholders*, como os usuários, os desenvolvedores, os vendedores, os gerentes, entre outros. Estes diferentes *stakeholders* podem perceber o valor de cada requisito a ser implementado em cada versão do software de formas distintas (FINKELSTEIN et al., 2009).

Quando se lida com múltiplos *stakeholders*, que podem concordar ou discordar sobre o valor a ser associado a cada requisito, a abordagem comum consiste em agregar múltiplas estimativas de valor em um único valor médio ponderado. Ao realizar tal agregação, informações importantes sobre a variabilidade das estimativas de valor dos requisitos são descartadas. Eventualmente, uma solução pode ser selecionada e assumida como ótima em termos da combinação custo-valor, porém esta mesma solução pode estar associada com uma ampla variação nos valores percebidos pelos diferentes *stakeholders* em relação aos seus requisitos (KHEIRKHAH; DERAMAN, 2008; AGARWAL; RATHOD, 2006; TONELLA; SUSI; PALMA, 2013). Tal variabilidade indica um alto grau de incerteza e, de forma correspondente, um elevado risco de insatisfação dos *stakeholders* (WATERIDGE, 1998; FAZIO et al., 1995). A redução ou atenuação do risco de insatisfação dos *stakeholders* é o núcleo da proposta desse trabalho de Tese.

1.1 O ESTADO DA ARTE

Uma questão importante que deve ser levada em consideração pelas companhias de software é a determinação de quais requisitos devem ser incluídos no próximo incremento do software, de forma que os *stakeholders* fiquem satisfeitos e haja um custo mínimo para a empresa (DURILLO et al., 2009). Essa situação é conhecida como problema da próxima versão do software, comumente referenciado na comunidade científica como “*The Next Release Problem*” (NRP) (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001).

O problema de seleção de requisitos é um problema combinatório, ou seja, são formados grupos de requisitos que respeitem as dependências técnicas existentes entre os requisitos e estes grupos são comparados a fim de se encontrar a combinação de requisitos que possa ser implementada dentro da restrição de custo disponível para a próxima

versão e que melhor atenda às demandas dos *stakeholders*. O espaço de soluções do NRP cresce de forma exponencial ao número de requisitos disponíveis para seleção, uma vez que uma possível solução é uma combinação de requisitos. Tratando-se de um problema de otimização, ele pode ser descrito como um problema de decisão.

Várias abordagens foram propostas na literatura para a seleção de requisitos em projetos de desenvolvimento de software (HARMAN; JONES, 2001), como *Analytic Hierarchy Process* (AHP) (SAATY, 1980), *Cost-Value* (KARLSSON; RYAN, 1997), *B-tree* (HEGER, 2004) e outros métodos tradicionais (AASEM et al., 2010; FELLOWS; HOOKS, 1998; BOEHM; GRÜNBAACHER; BRIGGS, 2001; WIEGERS, 1999). Algumas dessas abordagens usam técnicas que podem ajudar a encontrar um melhor conjunto de requisitos de software, de acordo com um conjunto de objetivos e restrições. No entanto, quando o espaço de busca é muito grande devido à presença de um número elevado de requisitos, essas técnicas falham ou apresentam um tempo computacional inaceitável (HARMAN, 2007; ZHANG; FINKELSTEIN; HARMAN, 2008; DEB; KALYANMOY, 2001).

Devido a esse problema, recentemente técnicas não-exaustivas e automatizadas têm emergido como uma estratégia alternativa para se encontrar soluções para o problema de seleção de requisitos de software quando as técnicas tradicionais não apresentam um resultado satisfatório (COELLO, 2000, 2000; DEB; KALYANMOY, 2001). Essas técnicas fazem uso de algoritmos probabilísticos, tais como algoritmos genéticos e meta-heurísticas (HARMAN; JONES, 2001; TALBI, 2009), e são amplamente aplicadas a problemas da Engenharia de Software, como o NRP (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001; HARMAN et al., 2012; PITANGUEIRA et al., 2013). Essa combinação entre técnicas probabilísticas aplicadas a problemas de busca na engenharia de software é conhecida como *Search Based Software Engineering* (SBSE) (HARMAN; JONES, 2001).

O NRP torna-se ainda mais complexo quando consideramos que esse problema tipicamente envolve dois ou mais objetivos (pelo menos custo e valor). A formulação mais comumente utilizada quando mais de um objetivo deve ser levado em consideração é conhecida como *Multiple Objective Next Release Problem* (MONRP) (ZHANG; HARMAN; MANSOURI, 2007). Nesta formulação, cada possível solução para o problema é avaliada levando-se em conta os vários objetivos que devem ser otimizados ao mesmo tempo (multi-objetivo). Como resultado destas avaliações, duas soluções podem se tornar incomparáveis se a primeira for melhor em um objetivo que a segunda e a segunda, por sua vez, for melhor que a primeira em outro objetivo.

1.2 PROBLEMATIZAÇÃO

Para se ter uma noção da complexidade do processo de seleção de requisitos de software na presença de múltiplos *stakeholders*, considere um cenário com apenas dois *stakeholders* e duas soluções candidatas a serem selecionadas que são aproximadamente equivalentes em termos de valor médio e o custo de desenvolvimento. Uma dessas soluções pode incluir requisitos que têm aproximadamente o mesmo valor estimado por todos *stakeholders*, enquanto a outra pode ter alguns requisitos que têm alto valor para o primeiro *stakeholder* e outros requisitos que têm valor alto para o segundo *stakeholder*. Enquanto a primeira solução provavelmente é valiosa para ambos *stakeholders*, a segunda é exposta a um maior

risco de insatisfação, porque há menos concordância e mais incertezas no valor percebido de alguns requisitos.

Quando esse cenário é expandido para um número elevado de *stakeholders*, e considerando que em sucessivos incrementos do software alguns *stakeholders* percebem que seus desejos, aspirações e opiniões não estão sendo considerados, há a possibilidade de que diversos deles estejam insatisfeitos em suas demandas. A consequência imediata disso pode conduzir a um dos problemas mais críticos para o projeto de software: a falta de envolvimento e comprometimento dos *stakeholders* durante todo o processo de desenvolvimento (HOODAT; RASHIDI, 2009; MELI, 1999; HAN; HUANG, 2007). De acordo com o *Standish Group* na descrição do *The Chaos Report*, a falta de envolvimento dos *stakeholders*, somada à falta de suporte executivo e expectativas não realistas, respondem por cerca de 31,6% de paralisações, atrasos ou cancelamentos de projetos de software.

Para ilustrar essa situação, considere-se o projeto de um software que possua como *stakeholders* quatro representantes de vendas, um diretor de tecnologia, um gerente de produtos, um líder da arquitetura de software e um líder de desenvolvimento. Cada um desses *stakeholders* possui um nível de importância para a empresa de software e, cada um deles atribui para cada requisito um valor que expressa a importância (ou desejo) desse requisito. Além disso, cada requisito de software também possui um custo associado. Logo, uma possível solução é um subconjunto de requisitos de software com um custo total e valor total.

No momento da seleção do conjunto de requisitos que comporá a próxima versão do software, é possível que exista uma solução com o mesmo valor estimado por todos os *stakeholders* enquanto uma segunda solução possua alto valor para os líderes, gerente e diretor, porém baixo valor para os representantes de vendas. Logo, essa segunda solução está mais exposta ao risco de insatisfação de alguns *stakeholders* (representantes de vendas) devido ao menor grau de concordância do valor percebido sobre o conjunto de requisitos dessa solução, além de suas aspirações não estarem sendo levadas em consideração na seleção dos requisitos de software.

Alguns trabalhos consideram o risco no problema de seleção de requisitos de software (COLARES et al., 2009; BRASIL et al., 2012; RUHE; TN; GREER, 2003; LI; HARMAN; LETIER, 2014), porém não levam em conta o risco associado à insatisfação dos *stakeholders*, ou seja, o risco de selecionar um conjunto de requisitos que é importante para alguns *stakeholders* e insignificante (ou até mesmo indesejável) para outros. Como descrito anteriormente, o risco de insatisfação dos *stakeholders* é um dos fatores que podem levar ao insucesso do projeto de software. Portanto, presume-se que esse tipo de risco deve ser sempre levado em consideração nas formulações multi-objetivo para o problema da próxima versão do software.

A partir do problema exposto acima, este trabalho de Tese introduz uma nova noção de risco, denominada *risco de insatisfação dos stakeholders*, que deve ser levada em conta ao selecionar uma solução para o MONRP no espaço de custo/valor dos requisitos. O risco de insatisfação dos *stakeholders* é abordado como o risco associado à incerteza das estimativas do valor de cada requisito realizadas por diferentes *stakeholders*. Uma vez que tal risco é maior quando existem grandes flutuações nos valores atribuídos pelos múltiplos *stakeholders* a cada requisito de software, esse risco é medido como a variância

das estimativas de valor feitas por diferentes *stakeholders* para cada requisito de software. Para abordar esse problema, o risco de insatisfação dos *stakeholders* é integrado a uma formulação multi-objetivo (*RA-MONRP* - *Risk Aware Multi-Objective Next Release Problem*) e técnicas baseadas em *Satisfiability Modulo Theory* (SMT) *solvers* são aplicadas para encontrar soluções para o RA-MONRP.

Para um entendimento do trabalho de Tese, a Figura 1.1 mostra uma visão geral do RA-MONRP. Considere um cenário de um projeto de software com múltiplos *stakeholders* e um conjunto de requisitos de software. Cada *stakeholder* possui um nível de importância para a companhia e atribui para cada requisito um valor que expressa a importância do requisito para o *stakeholder*. Além disso, cada requisito possui um custo associado e um conjunto de relações com outros requisitos (por exemplo, as relações de dependência) que devem ser satisfeitas para que o requisito seja implementado. Para a seleção dos requisitos que comporão a próxima versão do software, técnicas de busca são usadas para se obter os conjuntos de requisitos que apresentem as melhores relações entre custo e valor. O resultado do processo de busca é um conjunto de soluções e cada solução possui um subconjunto dos requisitos disponíveis para seleção.

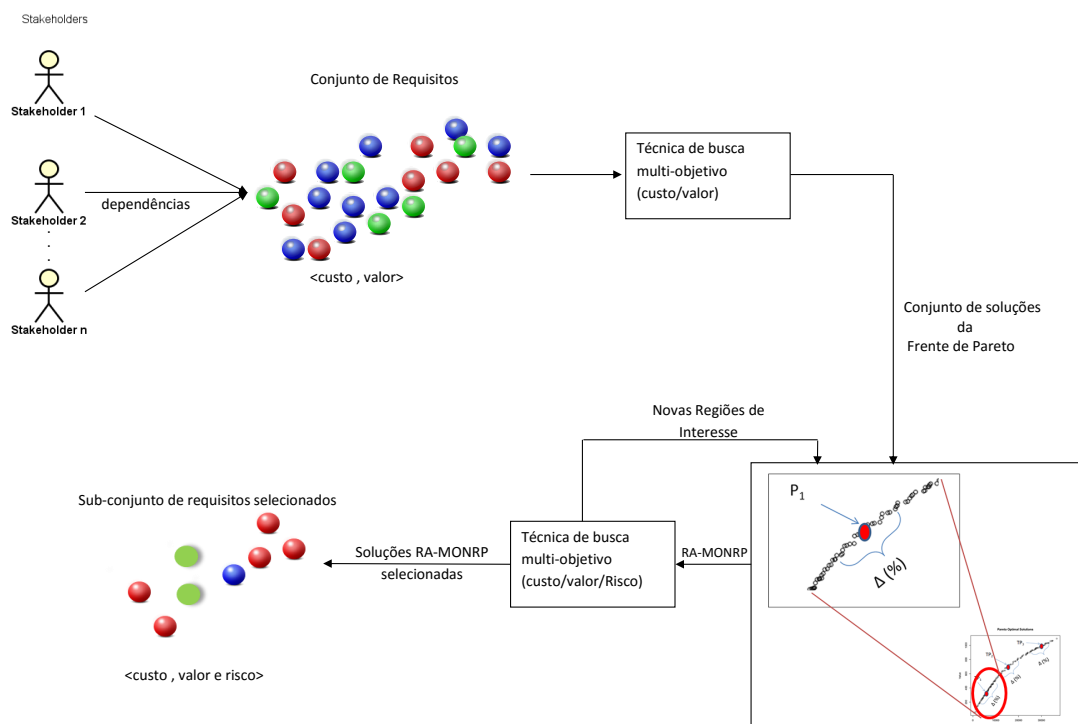


Figura 1.1: Visão Geral do RA-MONRP

No entanto, nesse momento uma solução pode ser escolhida agradando alguns *stakeholders* no que diz respeito ao custo e valor e, por outro lado, desagradando outros

nesses mesmos objetivos. Para abordar essa situação, os *stakeholders* podem escolher algumas soluções desse conjunto e especificar uma variação percentual aceitável para o custo e valor, estabelecendo assim uma vizinhança para cada solução (uma Região de Interesse, ou ROI). Cada região de interesse é incluída na formulação do RA-MONRP, onde são considerados o valor dos requisitos, seu custo e o **risco** de insatisfação dos *stakeholders*. Em seguida, uma técnica de busca é novamente executada, visando minimizar custo e risco e maximizar o valor dentro das regiões de interesse. O RA-MONRP pode ser executado quantas vezes for necessário e as regiões de interesse podem ser expandidas ou limitadas até que os *stakeholders* encontrem a solução que minimize o custo, maximize o valor e possua menor risco, de acordo com as suas expectativas.

Por se caracterizar como um estudo exploratório, a metodologia proposta para esse trabalho de Tese teve como atividade inicial a condução de uma revisão e mapeamento sistemático sobre seleção de requisitos de software com foco em técnicas baseadas em buscas (PITANGUEIRA et al., 2013). A partir dos resultados dessa revisão, foi realizada uma análise detalhada das tendências e possibilidades de novos focos de pesquisa na área e foi percebido que havia espaço para se aprofundar sobre o estudo do risco, com especial interesse no risco de insatisfação dos *stakeholders*.

Com essa perspectiva, as etapas seguintes do método de pesquisa seguiram as proposições de experimentação em engenharia de software (WIERINGA, 2014; WOHLIN et al., 2012) e da Engenharia de Software Baseada em Buscas (HARMAN et al., 2012). Inicialmente foi desenvolvida uma nova formulação para o risco, denominada RA-MONRP, e elaboradas as diretrizes para o processo de experimentação visando identificar as questões de pesquisa a serem respondidas. Feito isto, todo o planejamento dos experimentos para o RA-MONRP foi delineado e, posteriormente, deu-se a fase de execução dos experimentos. Por fim, os resultados obtidos foram catalogados, organizados, analisados e discutidos.

A proposta central desse trabalho visa desenvolver uma abordagem centrada na redução do risco associado à insatisfação dos *stakeholders* com mínimo impacto no custo e no valor agregado por um conjunto de requisitos, além de estudar a aplicabilidade das técnicas em diferentes cenários de escalabilidade.

1.3 OBJETIVOS E CONTRIBUIÇÕES

O objetivo geral desta Tese é propor uma nova formulação multi-objetivo para o problema da próxima versão do software para minimizar o risco de insatisfação dos *stakeholders* durante o processo de seleção de requisitos em um projeto de desenvolvimento de software. Para este fim, é formalizada uma abordagem multi-objetivo levando em consideração o risco associado às estimativas de valor dos requisitos de software e técnicas de busca são implementadas para otimizar o processo decisório quanto às soluções que possuam o menor risco e impliquem no mínimo impacto no custo e valor médio dos requisitos associados. Como objetivos específicos deste trabalho podem ser citados:

1. Realizar uma nova formalização multi-objetivo (RA-MONRP) para a inclusão do risco de insatisfação dos *stakeholders* no MONRP;
2. Implementar uma abordagem para a busca de soluções para o RA-MONRP baseada

na *Satisfiability Modulo Theory*;

3. Realizar estudos experimentais com o propósito de avaliar e validar a técnica de busca implementada;
4. Apresentar um estudo comparativo entre técnicas de busca exatas e meta-heurística desenvolvidas para o RA-MONRP.

Os objetivos e resultados desse trabalho de Tese contribuem para um avanço no estado da arte através de uma nova abordagem para a inclusão do risco de insatisfação dos *stakeholders*, além de um estudo comparativo da qualidade entre a aplicação de um método probabilístico (meta-heurística baseada em algoritmos genéticos) e um método determinístico (SMT) no problema MONRP.

As contribuições podem ser elencadas tanto para a área científica quanto para a indústria de software:

1. Uma revisão e mapeamento da literatura mostrando o cenário da seleção de requisitos de software com a utilização de técnicas baseadas em busca. O protocolo desenvolvido pode ser replicado para o estudo contínuo da área e atualização dos trabalhos científicos estudados;
2. Desenvolvimento de uma nova abordagem para a seleção de requisitos de software, com o objetivo de minimizar a insatisfação dos *stakeholders* ao longo do desenvolvimento de software. A abordagem proposta auxilia os profissionais de software a atenuar um dos maiores problemas de insucesso em projetos de software, que é o desinteresse dos *stakeholders* quando suas opiniões e desejos não são levados em consideração ao longo do projeto;
3. Desenvolvimento e avaliação de uma estratégia para obtenção de soluções através da seleção de regiões de interesse dos *stakeholders* e do uso de técnicas focadas em *Satisfiability Modulo Theory*, fundamentadas na área de pesquisa operacional, porém pouco aplicadas pela comunidade acadêmica de Engenharia de Software baseada em Buscas;
4. Demonstração de um conjunto de cenários de escalabilidade, em que técnicas exatas ou meta-heurísticas podem ser aplicados em problemas de seleção de requisitos de software. O teste de escalabilidade mostra o alcance dos dois tipos de técnicas e como elas se comportam em cenários com várias restrições impostas;
5. Estudo comparativo de qualidade das soluções encontradas por técnicas exatas e probabilísticas, possibilitando uma melhor compreensão da efetividade dessas técnicas probabilísticas no processo de busca e seleção de requisitos de software.

1.4 METODOLOGIA DA TESE

A metodologia usada nessa Tese está no contexto de métodos de pesquisa empíricos, em que um modelo é proposto e avaliado através de estudos empíricos, como por exemplo, experimentos controlados (BASILI, 1993). De forma mais detalhada, a metodologia usada nesse trabalho possui natureza exploratória, dado que os achados da pesquisa emergem das observações realizadas (WOHLIN et al., 2012); usa abordagem quantitativa, pois os resultados são passíveis de serem mensurados (LAKATOS; MARCONI, 2003) e se baseia em um procedimento experimental no qual as variáveis que atuam no fenômeno estudado são levantadas, delimitadas e controladas (KÖCHE, 2011).

O primeiro procedimento realizado foi a formulação da questão de pesquisa geral da Tese, a seguir: *“É possível desenvolver uma formulação para o risco de insatisfação dos stakeholders no processo decisório de seleção de requisitos de software?”*. Para responder a essa questão de pesquisa, uma série de etapas que compuseram a metodologia da Tese foram realizadas. A Figura 1.2 mostra essas etapas. Nela, é possível identificar tanto as atividades desenvolvidas quanto as publicações (P) realizadas ao longo da Tese, dispostas em ordem cronológica. A validação da Tese é baseada na diretrizes de Wohlin et al. (WOHLIN et al., 2012).

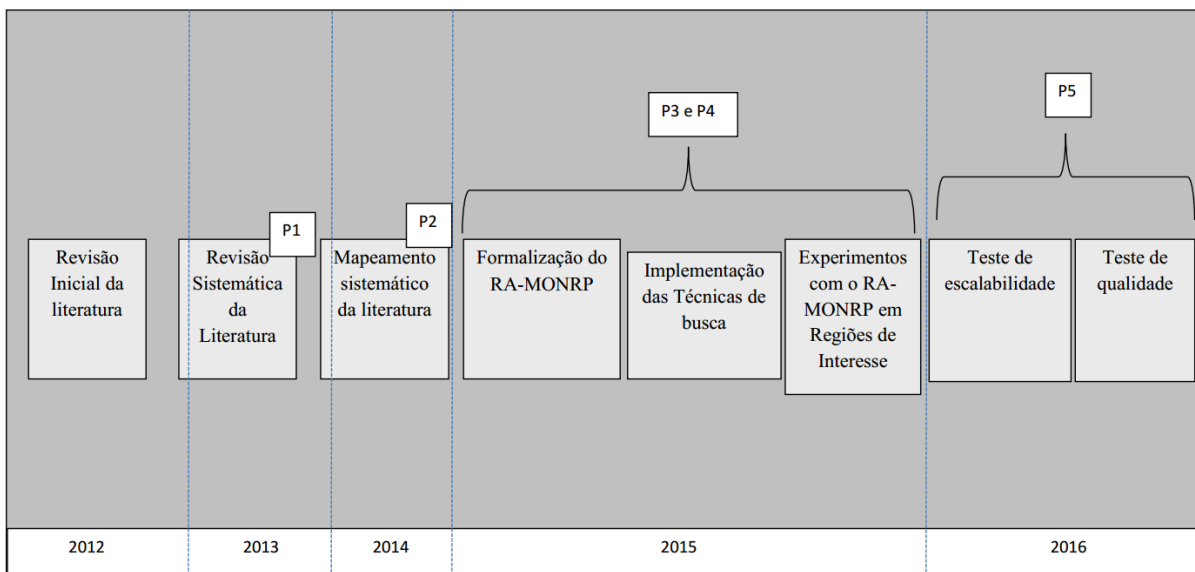


Figura 1.2: Metodologia da Tese

A primeira etapa foi uma **Revisão Inicial da Literatura** que foi executada no início do Doutorado. Nesta atividade realizou-se uma revisão detalhada, porém informal, da literatura que trata da seleção de requisitos de software e Engenharia de Software baseada em buscas, com especial interesse em estudos que levassem o risco em consideração nas suas propostas. A partir dessa investigação percebeu-se que alguns estudos incluíam o conceito de risco, mas esse não era o foco principal da pesquisa. Além disso, quantitativamente poucos trabalhos eram focados no risco, o que limitava a percepção de quais técnicas eram utilizadas em seleção de requisitos de software e o uso de diferentes

formulações para isto.

Através da percepção dessa primeira revisão, decidiu-se ampliar o escopo da investigação, não limitando-o apenas ao conceito de risco, mas levando em consideração uma perspectiva mais dilatada da seleção de requisitos de software baseada em busca, com a meta de se obter uma visão geral de todas as formulações, quais métodos de pesquisa são utilizados, quais técnicas de busca são aplicadas, como o processo de experimentação é desenvolvido e quais tendências e oportunidades existem nessa área. A busca por essa percepção levou à segunda atividade da metodologia, ou seja, o desenvolvimento de um protocolo detalhado para a confecção de uma **Revisão Sistemática da Literatura** sobre o processo de seleção de requisitos de software usando abordagens baseadas em busca. Esta revisão, naquele momento, era algo ainda não realizado na área. Através da revisão sistemática foi possível identificar o estado da arte em seleção de requisitos de software usando abordagens baseada em buscas.

Com os resultados da revisão sistemática, houve a necessidade de se fazer um **Mapeamento Sistemático** da área estudada com o objetivo de categorizar e classificar os estudos selecionados. Esse mapeamento auxiliou na visualização do resumo quantitativo geral da área. Tanto os resultados da revisão sistemática quanto do mapeamento sistemático são mostrados no capítulo 2 (Seção 2.2.1). Após esses dois trabalhos de revisão da literatura foi obtido um panorama geral e detalhado das possibilidades de trabalho científico na área de seleção de requisitos de software. Os aspectos já bem fundamentados na área, como também as possibilidades futuras, foram analisados e debatidos com o intuito de moldar o trabalho da Tese e servir como direcionamento para definição do problema a ser abordado.

Com esse cenário, percebeu-se que a inclusão do risco no processo de seleção de requisitos de software era algo pouco estudado e, mais especificamente, o risco de insatisfação dos *stakeholders* no processo de seleção de requisitos de software era algo novo a ser explorado. É importante salientar que, durante as atividades após a revisão e mapeamento sistemático, a técnica *snowball* (BIERNACKI; WALDORF, 1981) foi utilizada para atualizar constantemente os estudos dessa revisão, principalmente os trabalhos que abordassem o conceito de risco na seleção de requisitos de software. Portanto, com a revisão sistemática foi possível se ter uma base para a escolha das técnicas de busca e o *design* dos experimentos.

No ano de 2015 (vide Figura 1.2), foi realizado um estágio sanduíche na Fondazione Bruno Kessler (FBK), sob orientação do Dr. Paolo Tonella e Dr. Angelo Susi. Esse estágio foi crucial para o amadurecimento e evolução do trabalho realizado, uma vez que as três etapas essenciais para o trabalho de Tese foram realizadas.

A primeira delas foi a **formulação do RA-MONRP** como um problema de busca. Nessa etapa, o risco de insatisfação dos *stakeholders* foi formulado matematicamente e um modelo multi-objetivo foi confeccionado para servir de diretriz ao processo de experimentação (para maiores detalhes ver seção 5).

Testes iniciais com conjuntos de dados artificiais e um algoritmo exaustivo foram realizados para verificar a aplicabilidade, estudar os resultados para essa formulação e checar o alcance do algoritmo implementado em relação ao número de requisitos que poderiam ser analisados. Como os resultados foram satisfatórios mas o algoritmo exaustivo funcio-

nava apenas para instâncias reduzidas (com menos de 15 requisitos), decidiu-se selecionar novas técnicas de busca para usos em instâncias maiores. Essas técnicas de busca foram selecionadas através dos resultados da revisão e mapeamento sistemático e da atualização constante através do método *snowball*.

Feito isto, conjuntos de dados reais foram obtidos e seguiu-se com a **implementação das técnicas de busca** que serviriam de suporte ao processo de otimização do RA-MONRP (ver seção 5.2). Com as técnicas já implementadas, a próxima etapa da metodologia da Tese foi a realização de **experimentos com o RA-MONRP** em regiões de interesse do espaço de busca.

Em adição a estas etapas anteriores, durante o estágio doutoral duas novas atividades foram identificadas e especificadas para serem conduzidas: **testes de escalabilidade** (seção 6.3.2) e **testes de qualidade** (seção 6.3.3) foram aplicados e seus resultados analisados, formando assim, o conjunto total das atividades desse trabalho de Tese.

Conforme pode ser identificado na Figura 1.2, o trabalho dessa Tese gerou, até o momento, as seguintes publicações (P):

- (P1) PITANGUEIRA, A. M. et al. *A Systematic Review of Software Requirements Selection and Prioritization Using SBSE Approaches*, 5th International Symposium on Search-based Software Engineering (SSBSE 2013), pp. 188–208, St. Petersburg, Russia, August 24–26, 2013. Springer, Berlin, Heidelberg. ISBN 978-3-642-39742-4
- (P2) PITANGUEIRA, A. M.; MACIEL, R. S. P.; BARROS, M. *Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature*. Journal of Systems and Software, v. 103, pp. 267–280, 2015. ISSN 0164-1212.
- (P3) A. M. Pitangueira, *Incorporating preferences from multiple stakeholders in software requirements selection an interactive search-based approach*, 23rd IEEE International Requirements Engineering Conference (RE), Doctoral Consortium, Ottawa, ON, 2015, pp. 382–387. doi: 10.1109/RE.2015.7320455.
- (P4) PITANGUEIRA, A. M. et al. *Risk-Aware Multi-stakeholder Next Release Planning Using Multi-objective Optimization*. 22nd International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer International Publishing, Switzerland, LNCS 9619, 2016. pp. 1–15.

Além dos artigos acima, outro artigo foi submetido ao periódico *Information Software and Technology* e se encontra em processo de avaliação:

- (P5) PITANGUEIRA, A.M. et al. *Minimizing the Stakeholder Dissatisfaction Risk in Requirement Selection for Next Release Planning*.

1.5 ORGANIZAÇÃO DA TESE

Além dessa introdução do trabalho, essa Tese está organizada da seguinte forma. O Capítulo 2 apresenta o referencial teórico discutindo os conceitos principais que dão

suporte ao entendimento do estudo em questão. O Capítulo 3 apresenta uma revisão sistemática e mapeamento da área de seleção de requisitos de software com técnicas baseadas em busca. No capítulo 4 são apresentados os conceitos de riscos em projetos de software e os trabalhos relacionados a essa Tese. O Capítulo 5 mostra em detalhes a abordagem adotada na Tese e apresenta a formalização para o RA-MONRP. O Capítulo 6 refere-se a todo o procedimento de avaliação pondo em evidência o método de experimentação adotado, as questões de pesquisas a serem respondidas, os experimentos realizados e os resultados obtidos. Por fim, o Capítulo 7 mostra a conclusão e aponta os trabalhos futuros que serão realizados, de acordo com os resultados obtidos na Tese.

REFERENCIAL TEÓRICO

2.1 ENGENHARIA DE REQUISITOS

2.1.1 Requisitos de Software

Um produto de software deve satisfazer as necessidades de seus usuários, fornecedores, desenvolvedores e gerentes de projeto (KOTONYA; SOMMERVILLE, 1998). Para ser considerado de alta qualidade, este produto deve atender as necessidades, os desejos e as aspirações dos *stakeholders* (AGARWAL; RATHOD, 2006). Diversos fatores podem influenciar o desenvolvimento do produto de software, tais como, o investimento financeiro, habilidade técnica e engajamento da equipe envolvida no desenvolvimento do software, bem como a participação dos usuários nesse processo (KHEIRKHAH; DERAMAN, 2008). Nessa conjuntura, e de acordo com os gerentes de projeto, o cumprimento das exigências dos *stakeholders* é um dos critérios de sucesso mais importante para o produto de software (PFLEEGER, 2004).

As necessidades e exigências expostas pelos *stakeholders* para a construção de um software são transformadas em descrições dos serviços fornecidos pelo software e suas restrições operacionais, sendo entendidos como **requisitos** que esse software deve atender. (PRESSMAN, 2001).

De acordo com o *IEEE Standard Glossary of Software Engineering Terminology* (IEEE, 1990), um requisito é:

1. uma condição ou capacidade desejada por um usuário para resolver um problema ou alcançar um objetivo.
2. uma condição ou capacidade que deve ser atendida por um sistema ou componente do sistema para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos
3. uma representação documentada de uma condição ou capacidade como em (1) ou (2).

Conforme Hull et al. (HULL; JACKSON; DICK, 2005), requisitos são a base para todos os projetos, definindo o que os *stakeholders* necessitam em um potencial novo software e também o que o software deve fazer. O desafio maior é que os requisitos sejam bem entendidos por todos os *stakeholders* para capturar a necessidade ou problema de forma completa, inequívoca e sem ambiguidade.

No entanto, isso não é uma tarefa trivial, dado que as necessidades dos *stakeholders* podem ser muitas e variadas, e até mesmo conflitantes. Adicionalmente, estas necessidades podem não estar claramente definidas no início do projeto, podem ser limitadas por fatores fora do controle dos *stakeholders* ou podem sofrer mudanças no decorrer do ciclo de vida do projeto de desenvolvimento.

Como cada projeto de software possui características distintas e inerente ao domínio do projeto em questão, os requisitos existentes em cada projeto varia de contexto para contexto a depender da área de aplicação. Por exemplo, um requisito de software pode estar relacionado a automatização de parte de uma tarefa para alguém, visando apoiar os processos de negócio de uma organização, ou corrigir deficiências de software existente num dispositivo de controle (FAIRLEY; BOURQUE, 2014). Consequentemente, o entendimento do que cada requisito de software descreve é crucial no desenvolvimento, porque a falta de compreensão ou mau entendimento, aumenta o risco de não satisfazer as necessidades dos *stakeholders* (SIM; BROUSE, 2014).

Para uma melhor compreensão da importância dos requisitos de software em um projeto, alguns autores preferem categorizar os mesmos para detalhar o que cada requisito representa para um software (SOMMERVILLE, 2004; POHL, 2010; LOUCOPOULOS; KARAKOSTAS, 1995; PRESSMAN, 2001; HULL; JACKSON; DICK, 2005). Por exemplo, de acordo com Kandt (KANDT, 2003), requisitos de software podem ser divididos em técnicos ou não técnicos. Requisitos de software não técnicos incluem acordos, condições e termos contratuais. Exemplos destes são a identificação de marcos do projeto, produtos entregues, datas de entrega, e critérios de aceitação. Requisitos técnicos, por outro lado, são compreendidos como sendo funcional ou não funcional. Requisitos funcionais definem as capacidades funcionais, ou o comportamento de um produto de software. Os requisitos não funcionais identificam as restrições que um produto deve satisfazer, respeitando as qualidades obrigatórios do produto (ROBERTSON; ROBERTSON, 1999). Por exemplo, requisitos não funcionais, muitas vezes identificam as necessidades de utilização da CPU, dados, memória, operacional, desempenho, dimensionamento e sincronismo (KOTONYA; SOMMERVILLE, 1998).

Muitas outras categorizações podem ser encontradas (PRESSMAN, 2001; FAIRLEY; BOURQUE, 2014; LOUCOPOULOS; KARAKOSTAS, 1995), algumas vezes apresentando visões similares aos conceitos expostos acima. No entanto, Zhang e Harman (ZHANG; HARMAN, 2010) afirmam que, na prática, não há distinção clara entre as classificações de requisitos. É possível que um requisito pertença a várias categorias em termos de propriedades. Basicamente, a classificação do requisitos é uma maneira útil para investigar se os *stakeholders* compreenderam o que cada requisito de software especifica, tendo como objetivo a redução de problemas de entendimento advindos dos diferentes pontos de vista.

2.1.2 Fases da Engenharia de Requisitos

A engenharia de requisitos é uma abordagem sistemática e disciplinada para a especificação e gerenciamento de requisitos, tendo como metas: 1) Conhecer os requisitos relevantes, estabelecer um consenso entre os *stakeholders* a respeito de tais requisitos, documentá-los conforme os padrões estabelecidos pela empresa e gerenciá-los de forma sistemática; e 2) Compreender e documentar as expectativas e necessidades dos *stakeholders*, especificar e gerenciar os requisitos de forma a minimizar o risco de entregar um sistema que não atenda às suas expectativas e necessidades (POHL; RUPP, 2011).

Em um nível mais detalhado, de acordo com Lamsweerde (LAMSWEERDE, 2009), a engenharia de requisitos 1) identifica e analisa problemas de um software existente, 2) avalia objetivos, oportunidades e opções para o novo software, 3) define funcionalidades, restrições e responsabilidades e, 4) especifica e organiza todas as informações num documento de requisitos que deve ser mantido e atualizado por toda a evolução do software.

A engenharia de requisitos é a base de sustentação para todas as demais fases do desenvolvimento de software (LAMSWEERDE, 2009; POHL, 2010) pois dá suporte de forma recorrente às demais fases do ciclo de vida do software, visando a obtenção de um produto de software que obedeça o que cada requisito expressa ao longo do desenvolvimento do software.

Segundo Pohl (POHL, 1993), para que um software possa ser bem construído e atender as demandas dos *stakeholders* é preciso transformar conhecimento informal em representações, melhorar constantemente a compreensão inicial dos requisitos em uma especificação completa do software e direcionar os pontos de vista pessoais para um acordo comum da especificação dos requisitos.

A Figura 2.1, mostra em detalhes que quando há uma especificação completa, uma visão comum de acordo com as expectativas dos *stakeholders* e uma representação formal, o produto de software (representado como “Saída Desejada”) atende às aspirações dos *stakeholders*.

A dimensão denominada como especificação, lida com o grau de entendimento dos requisitos em um determinado tempo do desenvolvimento do software. Normalmente, no início do desenvolvimento, a concepção do que deva ser o software não é claramente compreensível (opaca). O grau de compreensão aumenta ao longo da evolução do software quando há uma identificação mais clara dos requisitos funcionais e não funcionais (POHL, 1993).

A dimensão que se refere a representação trata das diferentes categorias de representação usadas para expressar o conhecimento sobre o software, tais como, gráficos, sons, linguagem formal e informal, descrições por exemplos, animações, diagramas, linguagem natural, entre outros. Por fim, a última dimensão lida com o nível de concordância alcançada na fase de especificação.

De acordo com Pohl (POHL, 1993), no início do projeto de software há múltiplas visões pessoais do que é ou deva ser o software, sendo que a tendência é que o grau de concordância vá gradativamente crescendo ao longo da evolução do software devido a um melhor entendimento do que cada requisito especificado representa. Como o desenvolvimento de um software é um processo incremental, essas fases interagem constantemente

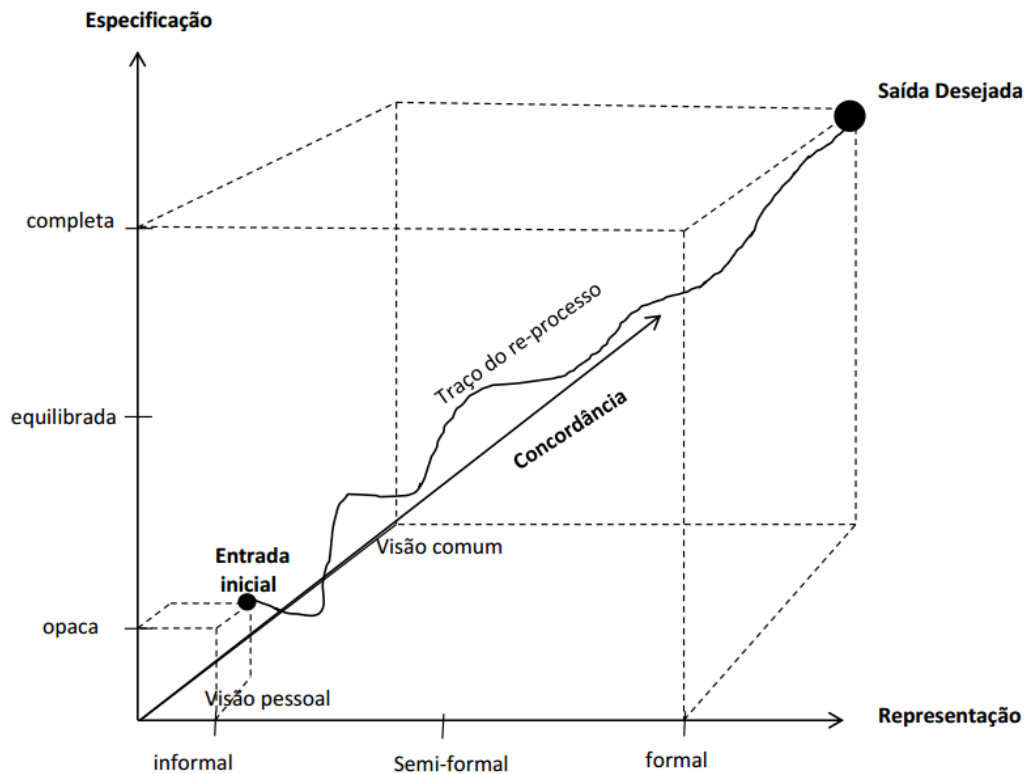


Figura 2.1: Três Dimensões da Engenharia de Requisitos. Adaptado de Pohl (POHL, 1993)

ao longo do processo, passando por diversos estágios no ciclo de desenvolvimento do software (LOUCOPOULOS; KARAKOSTAS, 1995).

As atividades intrínsecas dessas três dimensões citadas acima podem ser melhor detalhadas no clássico modelo evolutivo estudado por diversos autores (SOMMERVILLE, 2004; ZAVE, 1997; PFLEEGER, 2004; LI et al., 2007). A Figura 2.2 mostra um exemplo de visualização do modelo incremental. Nela, é possível identificar quatro principais atividades do processo de engenharia de requisitos: elicitacão, avaliação e acordos, especificação e documentação e validação e verificação. Essas etapas são importantes para o desenvolvimento do software pois são responsáveis por entender os desejos dos *stakeholders*, selecionar os requisitos de software que farão parte de cada versão do software (incremento) e analisar os possíveis riscos que possam ocorrer ao longo do ciclo de vida. (LAMSWEERDE, 2009; FAIRLEY; BOURQUE, 2014; POHL; RUPP, 2011; ZHANG; HARMAN, 2010)

A fase de elicitacão é o processo de identificação e consolidação das várias necessidades dos *stakeholders* (CHENG; ATLEE, 2007; KANDT, 2003). O propósito dessa fase é o ganho de conhecimento relevante do problema que pode ser usado para produzir uma especificação formal do software necessário para resolver o problema em questão a cada incremento (LOUCOPOULOS; KARAKOSTAS, 1995).

As tarefas principais dessa atividade, são 1) Identificar a razão para se desenvolver

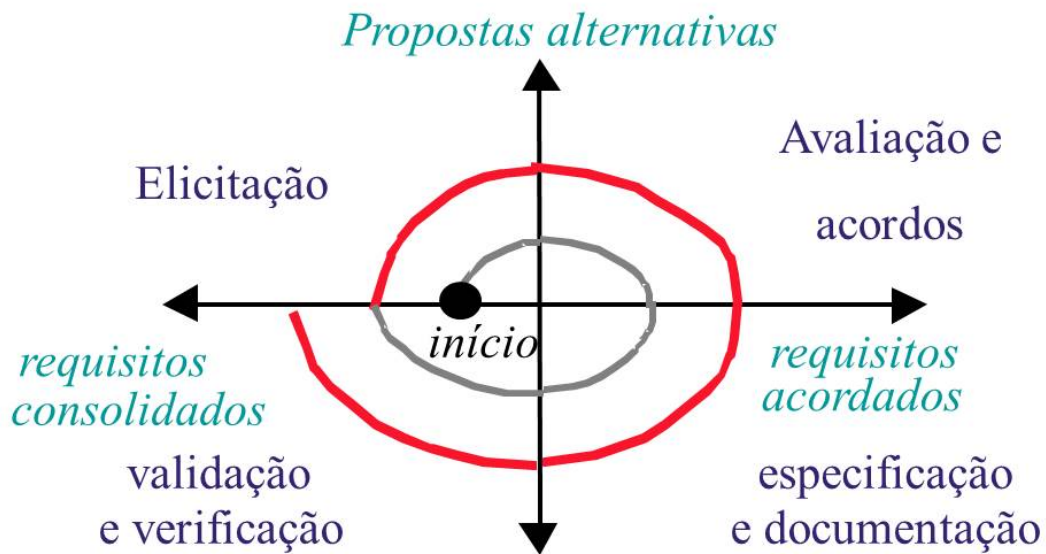


Figura 2.2: Processo incremental da Engenharia de Requisitos. Adaptado de Lamsweerde (LAMSWEERDE, 2009)

o software; 2) Identificar todas as fontes de conhecimento e envolver os *stakeholders*; 3) Definir uma visão clara e nítida do projeto de software; 4) Identificar os papéis dos usuários e suas características e 5) Identificar políticas operacionais aplicáveis ao contexto do projeto de software. (KANDT, 2003; LAMSWEERDE, 2009; LOUCOPOULOS; KARAKOSTAS, 1995).

Para cada incremento a ser desenvolvido, há uma etapa de avaliação e acordos que tem como meta tomar decisões sobre as questões levantadas na fase de elicitação visando estabelecer acordos sobre os pontos em pauta gerados por essa fase (LAMSWEERDE, 2009). Essa fase tem como principais atividades: a) a identificação e resolução de questões conflitante; b) identificação, discussão e avaliação dos riscos associados ao software proposto, c) comparação de opções alternativas levando em consideração os objetivos e riscos do projeto de software e a seleção das opções preferidas; e d) priorização de requisitos a serem selecionados para resolver conflitos, lidar com restrições de custo e tempo, e para dar suporte ao desenvolvimento incremental (POHL, 2010; LAMSWEERDE, 2009).

O produto final da avaliação tipicamente consiste numa proposta documentada contendo as decisões tomadas, após as avaliações e negociações realizadas (HULL; JACKSON; DICK, 2005). Essa proposta ressalta os requisitos e servirá como entrada para a próxima atividade - a especificação de requisitos (POHL; RUPP, 2011).

Uma especificação de requisitos deve estabelecer um entendimento entre os clientes e fornecedores sobre um produto que é suposto fazer, prover uma base para estimar os custos e cronograma e fornecer uma base para validação e verificação (IEEE98, 1998). Segundo Lamsweerde (LAMSWEERDE, 2009), essa fase do processo de engenharia de requisitos consiste em detalhar, estruturar e documentar as características do software a ser feito. As principais atividades da especificação de requisitos são: a) análise e assi-

milação do conhecimento sobre o que os requisitos expressam e b) síntese e organização desse conhecimento em modelo de requisitos coerente e formal (LOUCOPOULOS; KARAKOSTAS, 1995).

De acordo com Lamsweerde (LAMSWEERDE, 2009), o produto final gerado nessa fase é um Documento de Requisitos contendo os objetivos, conceitos, definições, propriedades do domínio a que pertence cada requisito e responsabilidades organizados numa estrutura coerente, que possa ser compreendida pelos *stakeholders*.

A fase de verificação e validação tem como propósito a garantia da qualidade do software. Para tanto, conforme Lamsweerde (LAMSWEERDE, 2009), as especificações resultantes da fase precedente devem ser cuidadosamente analisadas. Basicamente, essa fase visa certificar que as especificações dos requisitos de software estão consistentes e de acordo com as intenções dos *stakeholders* (LOUCOPOULOS; KARAKOSTAS, 1995). Portanto, as especificações dos requisitos devem ser validadas com os *stakeholders* a fim de encontrar inconsistências e omissões antes que os requisitos sejam transmitidos para os desenvolvedores do software (HULL; JACKSON; DICK, 2005; FAIRLEY; BOURQUE, 2014).

O produto da fase de validação e verificação é um modelo de requisitos que deve ser consistente e de acordo com as expectativas dos *stakeholders*. Conforme Loucopoulos e Vassilios (LOUCOPOULOS; KARAKOSTAS, 1995), na maioria dos casos, essa fase gera um compromisso entre o que foi desejado pelos *stakeholders* e o que é possível e viável sob as restrições de projeto. Um aspecto importante a ser ressaltado é que, apesar de toda formalização para se melhorar o processo de engenharia de requisitos e, por conseguinte, o desenvolvimento do software, há que se notar que essas atividades são altamente complexas e possuem aspectos de imprevisibilidade (HAN; HUANG, 2007). Tanto a complexidade quanto a imprevisibilidade fazem parte do processo de seleção de requisitos a cada incremento do software e os riscos inerentes ao projeto do software.

As próximas seções mostram em detalhes as características essenciais para a seleção de requisitos (seção 2.2) e os riscos (seção 4.2) que podem ocorrer durante um projeto de software.

2.2 SELEÇÃO DE REQUISITOS DE SOFTWARE

A atividade de selecionar requisitos é um problema complexo, porque envolve refinamentos, planejamento constante, avaliação e novas demandas dos *stakeholders* para cada incremento que comporá o produto final (SVAHNBERG et al., 2010).

Devido a esse cenário, o principal objetivo na seleção de requisitos de software é identificar as melhores escolhas para um conjunto de requisitos para satisfazer as exigências dos *stakeholders* (ZHANG; HARMAN; MANSOURI, 2007; HARMAN, 2007; DURILLO et al., 2009). Possíveis erros na seleção de requisitos de software podem ser extremamente caros e prejudiciais para o produto final, podendo levar a estouro de orçamento, insatisfação dos usuários e atrasos na entrega (FAIRLEY; BOURQUE, 2014; KANDT, 2003; HULL; JACKSON; DICK, 2005).

Várias abordagens têm sido propostas na literatura para apoiar a seleção de requisitos em projetos de desenvolvimento de software, tais como, *Analytical Hierarchy Process*

–AHP (SAATY, 1980), *Cost-Value* (KARLSSON; RYAN, 1997), *B-tree* (HEGER, 2004), *Numerical Assignments* (BERANDER; ANDREWS, 2005) e outros métodos tradicionais (AASEM et al., 2010; WIEGERS, 1999; FELLOWS; HOOKS, 1998; BOEHM; GRÜNBA-CHER; BRIGGS, 2001). Essas técnicas surgiram para auxiliar o tomador de decisão no processo de seleção em que a capacidade humana não consegue analisar, calcular e decidir sobre a melhor solução, dada a quantidade considerável de combinações possíveis a serem tratadas (KARLSSON et al., 1998; OCHOA, 2013; KUKREJA et al., 2012). Porém, com a demanda por software apresentando cada vez mais uma quantidade maior de requisitos, verificou-se que essas técnicas (consideradas exaustivas) diminuem consideravelmente o desempenho para os problemas atuais, que envolvem grandes espaços de busca (HARMAN; JONES, 2001; HARMAN, 2007; CARLSHAMRE et al., 2001).

Recentemente, técnicas de busca não exaustivas e com suporte automatizado surgiram como uma estratégia alternativa para resolver alguns problemas de seleção de requisitos de software que as técnicas exaustivas não atendiam (ABRAHAM; JAIN; GOLDBERG, 2005; BAKER; HARMAN; STEINH, 2006; TALBI, 2009). Muitas dessas técnicas advêm das propostas de otimização usadas pela pesquisa operacional (COELLO; LAMONT; VELDHUIZEN, 2007; COELLO, 2000; HILLIER; LIEBERMAN, 2004; IORDACHE; IORDACHE; MOLDOVEANU, 2014; WICKRAMASINGHE W; LI, 2008) e foram adaptadas aos problemas da engenharia de software.

A próxima seção detalha como a Engenharia de Software baseada em Buscas formaliza o problema de seleção de requisitos de software e discute as técnicas utilizadas para resolver esse problema.

2.2.1 Técnicas para a Seleção de Requisitos de Software Baseadas em Buscas

O termo *Search-based Software Engineering* (SBSE) foi introduzido na literatura em 2001, quando Harman e Jones aplicaram técnicas de busca em alguns problemas da área de engenharia de software (HARMAN; JONES, 2001). Técnicas de otimização já eram bastante usadas e difundidas em outras engenharias, principalmente, em problemas similares aos encontrados na engenharia de software, tais como, estimativa de custos de projetos, engenharia de requisitos, testes de sistemas, entre outros (COELLO; LAMONT; VELDHUIZEN, 2007; TALBI, 2009; DEB; KALYANMOY, 2001; ABRAHAM; JAIN; GOLDBERG, 2005; BRANKE et al., 2008).

O objetivo da área de SBSE é reformular problemas da engenharia de software como um problema de busca (HARMAN et al., 2009). Neste caso, entende-se um problema de busca como um tipo de problema em que soluções ótimas ou próximas destas, são procuradas dentro de um espaço de soluções candidatas, guiadas por uma função que distingue entre as melhores e piores soluções (HARMAN et al., 2012). Para tanto, tal processo de busca se dá através da modelagem (ou formalização matemática) do problema e, conforme Harman (HARMAN, 2007), são necessárias duas atividades para dar suporte a este processo, a saber:

1. A escolha da representação do problema;
2. Uma definição da função (ou funções) objetivo.

A primeira delas é essencial para a modelagem da natureza do problema que se está tentando solucionar porque determina qual o tipo de problema a ser abordado, as formas de abordá-lo, o(s) tipo(s) de objetivo(s) e quais possíveis técnicas podem ser aplicadas para se encontrar as soluções. Posteriormente, se define a função objetivo, sendo esta responsável pela caracterização do que pode ser minimizado ou maximizado. Esta função indica o panorama de solução desejado e direciona quais técnicas podem ser aplicadas neste panorama. Um dos maiores desafios da SBSE é então caracterizar o modelo de decisão para escolha da solução do cenário apresentado.

Diante desta caracterização inicial, Harman et al. (HARMAN et al., 2012) propuseram uma possível arquitetura para SBSE (ver Figura 2.3).

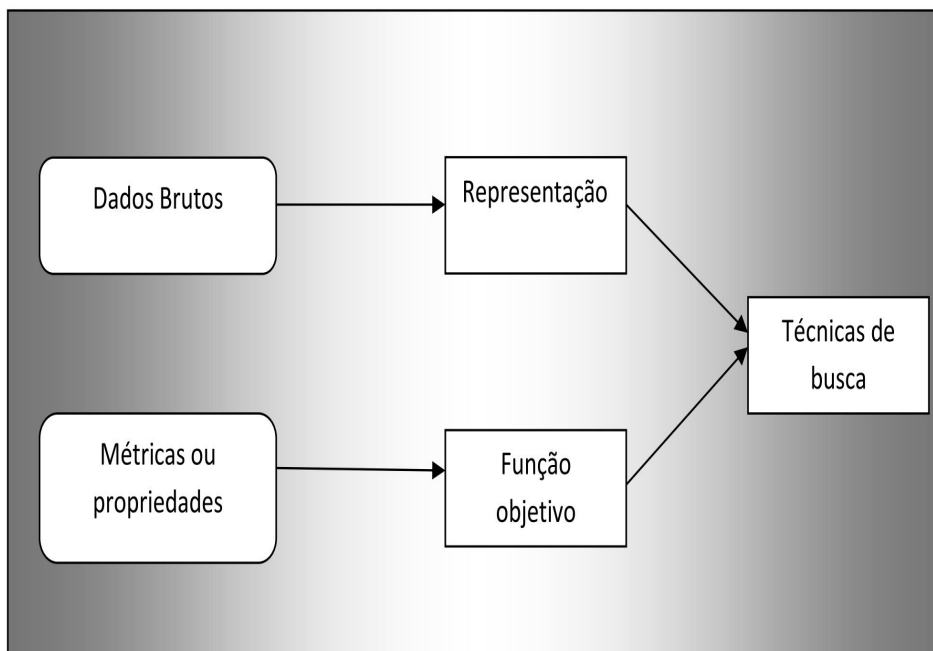


Figura 2.3: Arquitetura geral da Abordagem SBSE. Adaptado de Harman et al. (HARMAN et al., 2012)

Nessa arquitetura, os dados brutos são os elementos obtidos no contexto da aplicação que ainda não obtiveram nenhum tipo de tratamento computacional e que, portanto, precisam passar por uma compreensão do objetivo do problema para se ter uma representação adequada. Por exemplo, uma representação em grafo poderia ser escolhida para certos problemas, tais como as dependências entre os requisitos de software.

Além disto, precisa-se selecionar a métrica ou propriedade que mais se adéqua ao problema. Por exemplo, quais tipos de dependências entre requisitos serão levados em conta, métricas para a priorização dos requisitos, entre outros. De posse disso, a função objetivo é formalizada para verificar o objetivo determinado no problema (minimização e/ou maximização). Por fim, técnicas de busca seriam aplicadas para se obter os resultados.

De acordo com Zhang et al. (ZHANG; FINKELSTEIN; HARMAN, 2008; ZHANG et

al., 2011), as técnicas usadas em SBSE são bastante úteis para problemas na engenharia de requisitos, porque são capazes de auxiliar os *stakeholders* na tarefa complexa de selecionar requisitos de software, principalmente quando a capacidade humana e as técnicas exaustivas não conseguem explorar o espaço de busca quando há um complexo processo combinatorial.

A próxima seção mostra a aplicação de SBSE na engenharia de requisitos, mais especificamente, no problema de seleção de requisitos de software.

2.2.1.1 Problema da Próxima Versão A primeira abordagem usando SBSE para o problema de otimização na seleção de requisitos de software foi apresentada por Bagnall et al. (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001), que formulou o problema da próxima versão do software (NRP) como um problema de busca. Esse autor considerou que o desafio para a empresa desenvolvedora do software é selecionar uma série de requisitos dentro do limite de orçamento previsto e que, concomitantemente, atenda à demanda de seus clientes. Uma ilustração deste problema pode ser visto na Figura 2.4, onde é possível verificar a existência dos clientes e requisitos que eles desejam, e as dependências entre os requisitos.

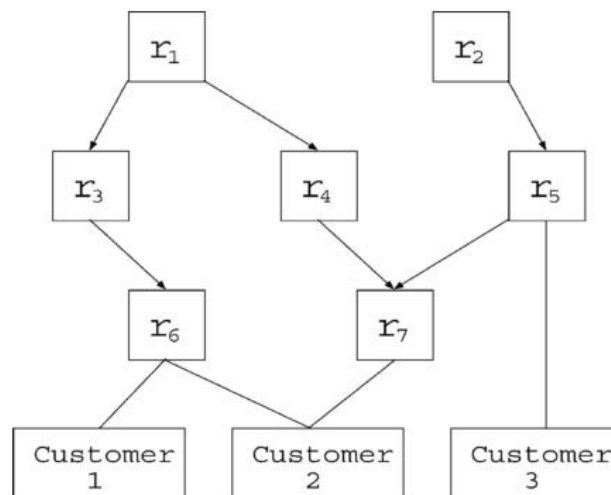


Figura 2.4: Exemplo da Estrutura do NRP. Adaptado de Bagnall et al. (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001)

Levando em consideração a Figura 2.4, a formulação original proposta por Bagnall et al. (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001) é um problema de maximização. O objetivo principal é maximizar o valor dos requisitos especificados pelos *stakeholders* sem que esse objetivo excedesse o orçamento disponível. Nessa formulação, existe um conjunto de n requisitos de software possíveis $R = \{R_1, \dots, R_n\}$ que possuem valores de importância atribuídos por um conjunto $S = \{S_1, \dots, S_m\}$ de *stakeholders*. Cada requisito R_i tem um custo associado, formando um vetor de custo $cost = [cost_1, \dots, cost_n]$. Cada stakeholder tem um grau de importância para a empresa de software, expressa como um conjunto de pesos relativos $W = \{w_1, \dots, w_m\}$ associado respectivamente com cada

stakeholder em S .

O valor de um requisito pode ser estimado diferentemente por cada stakeholder. Portanto, o valor que um requisito R_i tem para um stakeholder j é modelado como $value(R_i, S_j)$, onde um valor maior que 0 indica que o stakeholder S_j necessita (até certo ponto) do requisito R_i , enquanto um valor igual a zero indica o contrário (HARMAN et al., 2012). Sob estas premissas, a satisfação geral das partes interessadas para um determinado requisito é medido como uma média ponderada dos valores para todos os *stakeholders* $avgvalue_i = \sum_{j=1}^m w_j \cdot value(R_i, S_j)$, com $\sum_{j=1}^m w_j = 1$.

Uma solução para o NRP é um vetor $x = [x_1, x_2, \dots, x_n]$ representando um subconjunto de R , onde $x_i \in \{1, 0\}$, dependendo se R_i está incluído ou não na próxima versão do software. Além disso, restrições de dependências entre os requisitos são levadas em consideração, restringindo o espaço de solução admissível. A dependência entre R_i e R_j (resp. R_j e R_i) pode ser entendida como um par de requisitos $\langle R_i, R_j \rangle$ interpretado da seguinte forma: se R_j for incluído na próxima versão, R_i deve ser incluído também. Em outras palavras, a implicação $x_j \Rightarrow x_i$ deve ser atendida. O conjunto de dependências entre os requisitos é expressado como $D = \{\langle R_i, R_j \rangle, \dots\}$. Assim, o NRP pode ser formalizado como nas equações (2.1), onde B é o orçamento designado pela empresa para a próxima versão.

$$\begin{aligned} \text{Max } \sum_{i=1}^n x_i \cdot avgvalue_i & \quad \text{subject to } \sum_{i=1}^n cost_i \cdot x_i \leq B \\ & \quad \text{and } \bigwedge_{\langle R_i, R_j \rangle \in D} (x_j \Rightarrow x_i) \end{aligned} \quad (2.1)$$

A partir desta formulação, Zhang et al.(ZHANG; HARMAN; MANSOURI, 2007) elaboraram o MONRP, a versão multi-objetivo para o NRP. Para tanto, consideraram a maximização do valor para a empresa de software e a minimização do custo total necessário para implementar os requisitos selecionados. Esta formulação é apresentada nas equações (2.2). O MONRP não está limitado a dois objetivos, portanto, vários outros objetivos conflitantes podem ser adicionados à formulação, tais como a utilidade, risco, etc. (BRASIL et al., 2012; CAI; WEI, 2013; NGO-THE; RUHE, 2008).

$$\left\{ \begin{array}{l} \text{Min } \sum_{i=1}^n cost_i \cdot x_i \\ \text{Max } \sum_{i=1}^n avgval_i \cdot x_i \end{array} \right. \quad \text{subject to } \bigwedge_{\langle R_i, R_j \rangle \in D} (x_j \Rightarrow x_i) \quad (2.2)$$

De posse das formulações (NRP/MONRP) para a seleção de requisitos de software, a área de SBSE adota várias técnicas de busca para encontrar soluções para o problema exposto. A seção seguinte mostra e discute algumas dessas técnicas.

2.2.1.2 Técnicas baseadas em busca para o MONRP Para obter soluções no processo de seleção de requisitos de software de seleção por meio da abordagem SBSE, há uma necessidade de técnicas que auxiliem no processo de compreensão, análise e interpretação de resultados obtidos para a inclusão de requisitos para a próxima versão

do software. Nos casos mais simples (mono-objetivo), uma comparação equitativa entre alguns algoritmos (por exemplo, *Simulated Annealing* e *Hill Climbing*), observando a quantidade de esforço exigido por cada busca e uma análise estatística elementar, pode ser usada para analisar a melhor solução para um conjunto de requisitos (HARMAN et al., 2012).

No entanto, na prática, em um projeto de software é mais provável que existam muitos objetivos conflitantes para tratar quando for necessário determinar o conjunto de requisitos para incluir na próxima versão do software. Portanto, quando a meta de modelar o problema é a otimização de vários objetivos e restrições, há uma necessidade de técnicas para melhorar a busca de soluções, porque, ao contrário da otimização mono-objetivo (NRP), a solução de um problema multi-objetivo (MONRP) não é um ponto único com valor unitário, mas um conjunto de soluções expressando mais de um valor, de acordo com a quantidade de objetivos propostos (HARMAN et al., 2012; ZHANG et al., 2011; SAGRADO et al., 2010).

Esse conjunto de soluções é conhecido como Frente de Pareto, em que vários objetivos de otimização são combinados sem ter que decidir qual tem prevalência sobre os outros (HARMAN et al., 2012; HARMAN, 2007). Uma ilustração de uma Frente de Pareto pode ser vista na Figura 2.5.

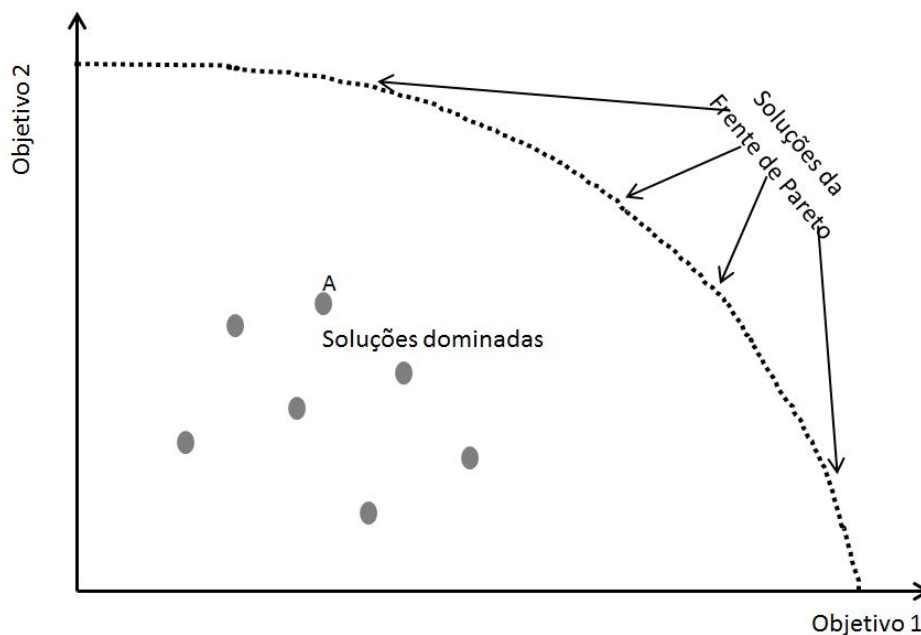


Figura 2.5: Exemplo de Frente de Pareto para problema de maximização de dois objetivos. Adaptado de Abraham et al. (ABRAHAM; JAIN; GOLDBERG, 2005)

Nessa Figura é possível perceber que o que diferencia as soluções da Frente de Pareto das demais é a dominância. De acordo com Deb (DEB; KALYANMOY, 2001), a dominância existe quando uma solução é melhor ou igual que uma outra solução em todos os critérios de otimização, e estritamente melhor em pelo menos um dos critérios.

Logo, as soluções fora da Frente de Pareto (por exemplo, Solução A na Figura 2.5) são completamente dominadas pelas soluções que estão na Frente de Pareto.

De forma mais detalhada, a otimização através da análise da Frente de Pareto é o processo de encontrar um conjunto de soluções não-dominadas pertencentes a um conjunto de soluções candidatas, e cada solução não dominada pode ser visto como um combinação ideal em todos os objetivos envolvidos na otimização (SAYYAD; MENZIES; AMMAR, 2013). Sendo assim, uma solução pertencente ao Pareto indica que não foi possível, no processo de busca, encontrar uma outra solução que melhora o desempenho de um dos objetivos a serem otimizados sem que diminua a qualidade de, pelo menos, um outro critério (TALBI, 2009).

Como muitas vezes, o espaço de busca para seleção de requisitos é imenso e há múltiplos objetivos conflitantes, métodos heurísticos são utilizados para se obter a Frente de Pareto. Conforme Hillier e Lieberman (HILLIER; LIEBERMAN, 2004), um método heurístico é um procedimento que, provavelmente, vai encontrar uma excelente solução viável, mas não necessariamente uma solução ótima, para um problema em questão. Não se pode dar garantia sobre a qualidade da solução obtida, porém uma heurística bem elaborada, em geral, é capaz de fornecer uma solução que se encontra pelo menos próxima da ótima (COELLO; LAMONT; VELDHUIZEN, 2007; TALBI, 2009; ABRAHAM; JAIN; GOLDBERG, 2005; EHRGOTT, 2000).

Um avanço em relação a essas técnicas são as meta-heurísticas. Uma meta-heurística é um método de resolução geral que fornece tanto uma estrutura quanto diretrizes de estratégia gerais para desenvolver uma solução. Na verdade, ela é um tipo de método que orquestra a interação entre procedimentos de melhoria local e estratégias de nível mais alto para criar um processo que seja capaz de realizar uma busca consistente em uma região de soluções viáveis (HILLIER; LIEBERMAN, 2004). Podem ser citadas como meta-heurísticas conhecidas: *Tabu Search* (GENDREAU; POTVIN, 2005), *Simulated Annealing* (KIRKPATRICK; GELATT; VECCHI, 1983) e, algoritmos genéticos (GENDREAU; POTVIN, 2010).

Os Algoritmos Genéticos (AG) são uma espécie de algoritmo de busca bio-inspirado com base na evolução do conjunto de indivíduos resultantes da seleção natural e genética naturais (POLI; LANGDON; MCPHEE, 2008; ABRAHAM; JAIN; GOLDBERG, 2005). De acordo com Talbi (TALBI, 2009), a um AG geralmente aplica-se um operador de recombinação de duas soluções que desempenham um papel importante, além de um operador de mutação que altera aleatoriamente o conteúdo do indivíduo (uma possível solução) para promover a diversidade. Algoritmos Genéticos vem sendo utilizados com bastante frequência pela comunidade de SBSE para resolver o NRP e/ou MONRP (HARMAN et al., 2012; PITANGUEIRA et al., 2013) pois tendem a ser, particularmente, eficientes na exploração de diversas partes da região de soluções viáveis, obtendo uma evolução gradual no sentido das melhores soluções do espaço de busca (ZHANG; HARMAN, 2010; HARMAN et al., 2012; ZHANG; HARMAN; LIM, 2013).

Na área de SBSE, um dos algoritmos genéticos mais comumente utilizado é o NSGA-II (*Non-Sorted Genetic Algorithm*) desenvolvido por Deb et al. (DEB et al., 2002), que é muitas vezes utilizado em problemas complexos que envolvem a seleção de requisitos (ZHANG; HARMAN, 2010; FINKELSTEIN et al., 2009; DURILLO et al., 2011). As

principais características citadas por Zhang e Harman (ZHANG; HARMAN, 2010) sobre o NSGA-II foram a capacidade deste algoritmo para obtenção de melhores soluções em instâncias de dados grandes e a diversidade das soluções apresentadas no processo de otimização.

Uma outra maneira de abordar problemas de seleção de requisitos de software é o uso de hibridização. De acordo com Harman et al. (HARMAN et al., 2012), hibridização de técnicas de busca como uma meta-heurística e uma técnica linear (ex: programação inteira (AKKER et al., 2005b; PURSHOUSE et al., 2014)) pode ser benéfico para injetar soluções em algoritmos evolutivos, a fim de acelerar o processo de encontrar soluções próximas das soluções ótimas. Conforme Abraham et al. (ABRAHAM; JAIN; GOLDBERG, 2005), algumas das razões para a aplicação de hibridização incluem: (1) melhorar o desempenho do algoritmo evolutivo, por exemplo, sua velocidade de convergência; (2) melhorar a qualidade de soluções obtidas por meio da execução do algoritmo evolutivo; e (3) incorporar o algoritmo evolutivo como parte de um sistema maior.

As técnicas de busca e as formulações apresentadas mostram as possibilidades de abordagens para o problema de seleção de requisitos de software, no entanto, para o trabalho de Tese é necessário um panorama mais detalhado das aplicações e formas de uso das técnicas de busca. Portanto, o próximo capítulo mostra uma revisão e mapeamentos sistemáticos da literatura relacionada a esses temas aqui mostrados, identificando as técnicas de busca mais utilizadas para a seleção de requisitos de software e as formulações usadas tanto para o NRP quanto para o MONRP.

2.2.2 Considerações Finais

Este capítulo apresentou o referencial teórico sobre os conceitos fundamentais utilizados nesse trabalho de Tese. A Engenharia de Requisitos e suas fases foram discutidas, bem como o processo de seleção de requisitos de software, onde foi possível apresentar os tipos de abordagem (NRP e MONRP) e algumas técnicas que são usadas no processo de busca para a seleção de requisitos de software.

O próximo capítulo apresenta uma revisão e mapeamento sistemático sobre seleção de requisitos de software, apresentando resultados sobre formas de se abordar o problema de seleção de requisitos de software, as técnicas de busca mais utilizadas nesse problema, as tendências da área e as possibilidades de novos estudos para o problema de seleção de requisitos de software.

REVISÃO E MAPEAMENTO SISTEMÁTICO SOBRE SELEÇÃO DE REQUISITOS DE SOFTWARE USANDO A ABORDAGEM SBSE

Realizamos uma revisão sistemática com o intuito de examinar o estado atual do processo de seleção de requisitos de software, com foco nas técnicas de engenharia de software baseada em buscas (PITANGUEIRA; MACIEL; BARROS, 2015). A revisão sistemática permite obter um panorama dessa área, os pontos fortes e fracos da seleção de requisitos de software usando abordagens SBSE, a identificação das tendências de investigação nessa área, e a observação dos campos de interesse específicos da comunidade acadêmica. Além da revisão sistemática, um mapeamento da literatura foi realizado com o intuito de categorizar, identificar e classificar os estudos selecionados, mostrando um resumo visual e quantitativo dos trabalhos previamente realizados na área (PITANGUEIRA; MACIEL; BARROS, 2015; PITANGUEIRA et al., 2013).

Uma revisão sistemática é uma técnica para identificar, avaliar e interpretar pesquisas relevantes em uma determinada área de interesse, uma pergunta de pesquisa ou um fenômeno específico de interesse (KITCHENHAM, 2004). Revisões sistemáticas fornecem um resumo conciso de evidências disponíveis sobre uma determinada área de interesse. Essa abordagem usa métodos rigorosos e explícitos para identificar, avaliar criticamente e sintetizar estudos relevantes sobre um tema específico (DYBA; DINGSOYR; HANSSSEN, 2007). Há várias razões para a realização de uma revisão sistemática (KITCHENHAM, 2004), como por exemplo, para resumir a evidência disponível sobre um tratamento de uma tecnologia; identificar lacunas na pesquisa atual, a fim de sugerir áreas para promover a nova pesquisa e fornecer uma estrutura para posicionar corretamente as novas atividades de pesquisa científica.

Uma revisão sistemática geralmente compreende as etapas de 1) planejamento (identificação da necessidade de uma revisão) e desenvolvimento de um protocolo de avaliação, 2) condução (identificação da pesquisa), seleção de estudos primários, estudo de avaliação da qualidade, extração e síntese de dados e elaboração de relatórios e 3) elaboração do relatório contendo todo o processo e os resultados.

De acordo com Petersen et al. (PETERSEN et al., 2008), um mapeamento sistemático é um método de investigação que categoriza resultados que foram publicados, fornecendo um resumo visual dos seus resultados e um relatório contendo os resultados da pesquisa. A análise é feita nas publicações em um campo de pesquisa e os resultados podem ser usados para responder a perguntas específicas de investigação. Em outros termos, pretende-se mostrar como os estudos são distribuídos na área de pesquisa selecionada. Os resultados provenientes do estudo do mapeamento subsidiam a questão de investigação formulada na revisão sistemática, mostrando tendências e quantidades específicas.

A próxima seção detalha toda a fase de planejamento e o processo de execução da revisão sistemática e mapeamento da área de estudo citada anteriormente.

3.1 PLANEJAMENTO E EXECUÇÃO DA REVISÃO SISTEMÁTICA

Uma revisão sistemática requer um protocolo bem especificado que descreve a dinâmica do processo (KITCHENHAM, 2004). Uma das atividades mais importantes na fase de planejamento é a formulação adequada de uma pergunta de partida ou questões específicas que devem ser respondidas pela revisão sistemática. Todas as outras fases do processo são dependentes desta formulação (DYBA; DINGSOYR; HANSSSEN, 2007).

A partir disso, a nossa pergunta de pesquisa (RQ) para esta revisão sistemática é:

Pergunta de Partida *Qual é o estado da arte para as aplicações de otimização baseada em buscas no processo de seleção de requisitos de software?*

Este é um tópico de importância científica e profissional, dado o crescimento contínuo de técnicas de busca para obtenção de soluções ótimas (ou próximas a estas) em problemas complexos, tal como é o de seleção de requisitos em projetos de software (ZHANG; FINKELSTEIN; HARMAN, 2008). Derivada da pergunta de partida, as questões de pesquisas foram divididas em questões associadas ao mapeamento (MQ) e questões relativas à revisão sistemática (RQ). As questões do mapeamento são as seguintes:

- **MQ1:** Como as publicações em seleção de requisitos de software usando abordagens SBSE estão distribuídas através dos anos?

Nesta questão, visou-se ter a noção do crescimento ou a redução da área de pesquisa.

- **MQ2:** Quem são os autores mais ativos e influentes na área?

Através desta pergunta, obteve-se uma indicação dos principais pesquisadores na área, indicando uma referência para publicações relacionadas às pesquisas futuras.

- **MQ3:** Quais fóruns de publicação são os principais alvos para a produção nessa área de pesquisa?

Nesta questão, pretendeu-se saber onde podem ser encontradas publicações sobre seleção de requisitos de software com enfoque em SBSE, bem como identificar bons alvos para publicações de estudos futuros.

Em relação à revisão sistemática, através de um estudo preliminar, as seguintes hipóteses e questões de pesquisas foram formuladas:

Hipótese 1: A quantidade de estudos usando o problema com formulação multi-objetivo é uma tendência recente e tende a aumentar quando comparada com o número de estudos realizados que usam a formulação mono-objetivo?

Esta hipótese motivou a construção da seguinte questão de pesquisa:

RQ1: Que tipos de modelagem (ou formulação) estão sendo usados na seleção de requisitos de software com foco em SBSE?

Nesta questão, buscou-se identificar como os pesquisadores modelam o problema de seleção de requisitos de software e como esse problema é abordado no que diz respeito à busca de soluções. Em outras palavras, como se formula o problema, as funções objetivos e as restrições que são adotadas na representação do modelo (CLARKE et al., 2003).

Hipótese 2: Técnicas de Meta-heurísticas são as mais utilizadas pelos pesquisadores para resolver o problema de seleção de requisitos de software.

De acordo com esta hipótese, foi formulada a seguinte pergunta de pesquisa:

RQ2: Quais técnicas de busca são utilizadas para a seleção de requisitos de software com foco em SBSE?

Relativo às técnicas (RQ2), estas podem ser entendidas como uma forma de se obter o conjunto de soluções para o problema, como por exemplo, uso de meta-heurísticas, aplicação de métodos híbridos ou um método de busca exata. Neste caso, a aplicação de técnicas advindas da área de pesquisa operacional podem ser usadas como uma solução para o problema da seleção de requisitos de software (HARMAN, 2007). De forma específica, a RQ2 visou saber o cenário do uso de técnicas para a seleção de requisitos de software.

Com base na hipótese acima, houve um interesse em saber a distribuição quantitativa das técnicas aplicadas na seleção de requisitos de software. Uma vez que há uma grande variedade numérica destas técnicas (COELLO; LAMONT; VELDHUIZEN, 2007; TALBI, 2009; ABRAHAM; JAIN; GOLDBERG, 2005; DEB; KALYANMOY, 2001), é essencial saber o cenário de uso na área de SBSE. Por conseguinte, formulou-se as seguintes hipótese e questão de pesquisa:

Hipótese 3. Na área de SBSE, um consenso entre o uso de técnicas de pesquisa não existe.

Como consequência, a questão de pesquisa sobre essa hipótese visa identificar o conjunto de técnicas usadas em SBSE. Esta é a pergunta:

RQ3. Quais técnicas de busca são utilizadas no processo de seleção de requisitos de software com foco em SBSE?

Em específico, o termo “técnica de busca” refere-se às maneiras de resolver o problema, observando os objetivos propostos, por exemplo, o uso de um algoritmo genético ou algoritmo exato.

Estas hipóteses e questões de pesquisa foram essenciais para determinar a estrutura e o conteúdo desta revisão, e também para orientar o processo, dado que todas as outras fases desta revisão sistemática e mapeamento derivam destas questões.

3.1.1 Identificação da Pesquisa - Estratégia de Busca

A fim de realizar uma pesquisa exaustiva para estudos primários, a estratégia de busca consistiu, inicialmente, em uma busca manual nas principais conferências e, posteriormente, em uma pesquisa *on-line* em bibliotecas digitais. Para garantir a qualidade desta revisão, procurou-se manualmente na conferência principal dessa área, *Symposium on Search-Based Software Engineering* (SSBSE) ¹ (2009 a 2013) e no repositório SBSE ².

Logo após, os seguintes bancos de dados eletrônicos foram pesquisados por serem considerados como os mais relevantes na busca de artigos científicos: *IEEE Xplore* ³, ACM Digital Library ⁴, ISI Web of Science ⁵, SpringerLink ⁶ e Science Direct ⁷. Além disso, também foram efetuadas buscas na *International Requirements Engineering Conference (RE Conference)* ⁸ e na *International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)* ⁹ por serem importantes conferências na área de engenharia de requisitos de software.

Ao usar a pesquisa *on-line* em bibliotecas digitais, o uso de palavras-chave é muito importante para obter bons resultados através da recuperação de informação. Basicamente, encontrar uma resposta (ou várias respostas) para uma questão de pesquisa inclui selecionar recursos de informações adequadas e executar uma estratégia de pesquisa detalhada. Então, com base na pergunta de partida e nas questões de pesquisa levantadas, foram

¹<http://ssbse.org/2016/>

²http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/repository.html

³<http://ieeexplore.ieee.org/Xplore/home.jsp>

⁴<http://dl.acm.org/>

⁵<http://thomsonreuters.com/thomson-reuters-web-of-science/>

⁶<http://link.springer.com/>

⁷<http://www.sciencedirect.com/>

⁸<http://requirements-engineering.org>

⁹<http://refsq.org/2014/past-conferencesz>

executadas buscas nas bases e conferências com critérios de pesquisa distintos, usando diferentes combinações de sequências de caracteres para não restringir em excesso o espaço de busca e nem criar um espaço infinito (KITCHENHAM, 2004; DYBA; DINGSOYR; HANSEN, 2007). As seguintes sequências de palavras-chave (ou termos de pesquisa) foram consideradas as mais apropriadas para a busca: 1) *Software requirements AND selection*; 2) *Software requirements AND prioritization*; 3) *Search Based AND requirements optimization*; 4) *Search Based AND requirements selection*; e 5) *Search Based AND requirements prioritization*.

Todos estes termos de pesquisa foram combinados usando o operador booleano “or”, que implica que um artigo tinha que incluir, pelo menos um dos termos para ser recuperados pelo motor de busca da base. Ou seja, a busca foi configurada da seguinte forma: item 1 ou 2 ou 3 ou 4 ou 5. Os termos de busca *software requirements*, *selection* e *prioritization* derivam-se dos termos da área específica. A inclusão do termo *Search-Based* foi devido a necessidade de restringir a pesquisa aos artigos que estavam na área de engenharia de software com a abordagem SBSE, para evitar a recuperação de trabalhos que incidiam sobre a seleção de requisitos de software, porém em campos de pesquisa diferentes de SBSE. Esses termos de pesquisa foram aplicados para os títulos, resumos e palavras-chave dos artigos em cada base de dados eletrônica exposta anteriormente.

Devido as diferentes funções e características dos motores de busca, houve a necessidade de implementar a estratégia de busca individualmente para cada banco de dados. Isto criou considerável trabalho extra, pois foi necessário realizar várias tentativas apenas para descobrir como cada um dos bancos de dados tratados julgava as diferentes expressões booleanas. A próxima seção explica o procedimento adotado de inclusão e exclusão de artigos no processo de busca.

3.1.2 Critérios e Procedimentos para a Seleção de Estudos

Os critérios de inclusão e exclusão foram definidos no protocolo de revisão que foi desenvolvido para esta revisão sistemática seguindo o roteiro adotado por Kitchenham (KITCHENHAM, 2004). Os estudos primários incluídos devem pertencer a uma das seguintes categorias:

- Categoria A: estudos propondo um novo modelo ou abordagem usando SBSE que podem ser aplicados em seleção de requisitos de software; ou
- Categoria B: estudos aplicando modelos existentes e técnicas para seleção de requisitos de software usando SBSE.

Os seguintes tipos de estudos foram excluídos durante a fase de seleção do estudo:

- Estudos sobre seleção de requisitos de software que não usam uma abordagem de Engenharia de Software baseada em buscas;
- Estudos sobre treinamento, editoriais, resumos, entrevistas, prefácios, notícias, comentários, correspondência, tutoriais, sessão de *pôster*, oficinas e painéis.

O processo de busca na revisão sistemática foi realizado entre os anos de 2012 e 2013; e nas restrições de busca, o ano de 2001 foi tomado como parâmetro, dado que, o termo SBSE foi criado nesse ano (HARMAN; JONES, 2001) e seus trabalhos categorizados a partir dessa data.

A Figura 3.1 a seguir detalha todos os estágios de busca e análise para inclusão ou exclusão dos estudos recuperados pelas buscas. Os estudos que fizeram parte do conjunto final da seleção foram avaliados por um conjunto de critérios de qualidade propostos por Barros e Dias-neto (BARROS; DIAS-NETO, 2011).

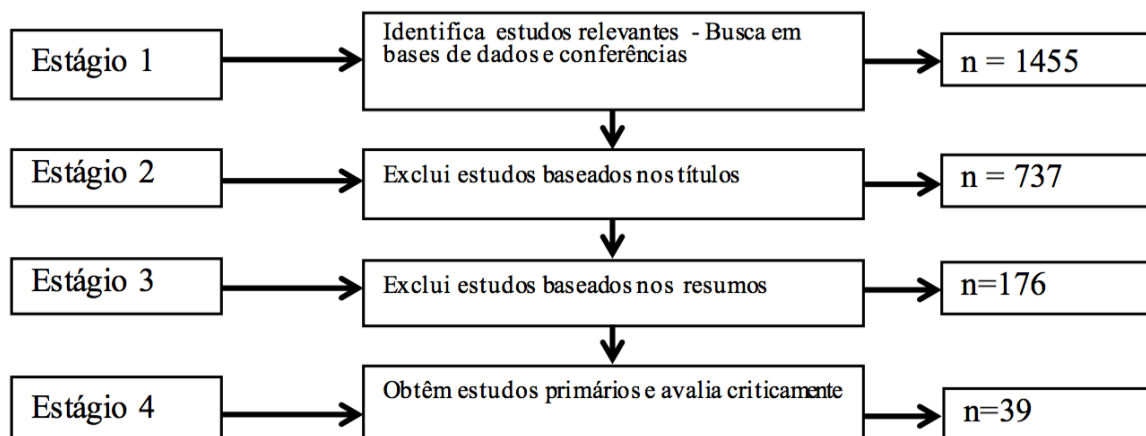


Figura 3.1: Estágios do processo de seleção de artigos (PITANGUEIRA; MACIEL; BARROS, 2015).

Como pode ser visto, a busca e seleção de estudos começou com a procura de estudos relevantes do assunto em questão, com base em seus títulos, resumos e palavras-chave. Estudos relevantes são potenciais candidatos para estudos primários (DYBA; DINGSOYR; HANSSEN, 2007) e os trabalhos completos foram totalmente analisados.

Cada estudo relevante foi analisado para determinar se era um estudo preliminar. Isto foi necessário porque na área de engenharia de requisitos, o título, resumo e palavras-chave não são geralmente suficientes para determinar o conteúdo de um artigo (LAMSWEERDE, 2009; CHENG; ATLEE, 2007). Portanto, cada estudo foi criteriosamente avaliado antes de se tomar uma decisão final sobre a possibilidade de incluir ou excluí-lo.

No estágio 1, citações duplicadas foram identificadas e removidas. No segundo estágio, estudos que claramente não eram sobre os requisitos de engenharia de software foram excluídos. Títulos que eram claramente fora do âmbito do assunto da revisão sistemática foram removidos. No entanto, em caso de dúvida, alguns documentos foram considerados para o terceiro estágio. Neste, estudos foram excluídos com base em seus resumos, caso seu objeto de estudo não era a seleção de requisitos de software usando a abordagem SBSE. No entanto, há que se ressaltar que alguns resumos apresentavam poucas informações (ou até informações incompletas), o que gerou um trabalho extra para identificação e categorização do artigo. Nesse estágio 3 foram incluídos todos os estudos que compartilhavam algumas características com a proposta de estudo dessa revisão sistemática.

Finalmente, no último estágio estudos foram elegíveis para inclusão na revisão sistemática se eles apresentassem todas as características citadas nas Categorias A e B. Como pode ser identificado na Figura 3.1, ao final de todo processo de busca e identificação de estudos apropriados, um total de 39 estudos foram selecionados para análise e avaliação. Cada um desses estudos foi avaliado com base no conjunto de critérios de abordagem sistemática para avaliação de estudos em SBSE propostos por Barros e Dias-neto (BARROS; DIAS-NETO, 2011). A próxima seção mostra cada um desses critérios e todo o processo de avaliação da qualidade dos estudos selecionados.

3.1.3 Avaliação da Qualidade dos estudos - Critérios adotados

A Tabela 3.1 apresenta a lista de critérios de qualidade adotados para a avaliação dos estudos selecionados. Como dito anteriormente, estes critérios foram baseados em um estudo anterior (BARROS; DIAS-NETO, 2011), em que os autores desenvolveram uma abordagem sistemática para avaliar estudos experimentais no campo de SBSE. A partir disso, foi desenvolvida uma lista de verificação em que cada pergunta/critérios tem um “Sim” ou “Não” como resposta.

Tabela 3.1: Critérios de qualidade selecionados para avaliação dos estudos (BARROS; DIAS-NETO, 2011)

Critérios de Qualidade
Validade Geral:
1- Existe uma introdução apropriada para a proposta do artigo?
2- O estudo apresenta uma ideia clara dos objetivos da pesquisa?
3- Há uma exposição clara dos resultados obtidos?
4- Existem limitações ou restrições aos resultados encontrados?
5- O estudo faz uma projeção de valor para a pesquisa e a prática?
Validade Interna:
6- O código fonte utilizado no estudo está discutido e disponibilizado para os demais pesquisadores?
7- O artigo detalha o procedimento de coleta dos dados utilizados no experimento?
8- O artigo apresenta uma definição explícita da instância de destino (aleatória e/ou real)?
Validade de Conclusão:
9- O experimento considera a variação aleatória (diversas execuções) nos algoritmos apresentados?
10- As hipóteses do estudo estão formalmente apresentadas?
11- Testes de inferência estatística são usadas?
12- Há uma base de comparação significativa entre a técnica utilizada e as melhores soluções conhecidas?
Validade de Construção:
13- O artigo discute o modelo adotado para o processo de otimização?
14- O artigo discute a validade das medidas de efetividade?
Validade Externa:
15- O estudo apresenta claramente a estratégia de seleção de instâncias (dados reais ou gerados aleatoriamente)?
16- O estudo trata a variação na complexidade das instâncias usadas no experimento?
17- O estudo usa instâncias do mundo real no experimento?
18- O estudo apresenta os valores dos parâmetros usados no experimento?

Para um melhor entendimento da avaliação, foi decidido dividir estes critérios em cinco grupos. O primeiro refere-se a aspectos gerais e informações estruturais sobre o artigo e visa avaliar se o trabalho selecionado está dentro do contexto de experiências na área de engenharia de software (JEDLITSCHKA; PFAHL, 2005; SJOBERG et al., 2005; KITCHENHAM et al., 2010).

Os demais grupos estão relacionados às ameaças à validade dos experimentos. Como proposto por Wohlin et al. (WOHLIN et al., 2012) as ameaças à validade são os riscos associados com o projeto, implementação e análise de resultados de estudos experimentais. Os critérios incluídos no grupo de validade interna são para avaliar se a relação é entre o tratamento observado e os resultados. Em outras palavras, o experimento deve garantir que é uma relação de causa e não o resultado de um fator sobre as quais o pesquisador não tem controle.

O grupo de validade de conclusão contém critérios que relacionam o tratamento com os resultados, portanto, procedimentos estatísticos devem ser utilizados para avaliar a significância dos resultados (BARROS; DIAS-NETO, 2011). O grupo de validade de construção contém critérios que se preocupam com a relação entre a teoria e a observação realizada (WOHLIN et al., 2012). Finalmente, o último conjunto de critérios lida com a validade externa, centrando-se sobre a generalização dos resultados observados em um experimento para uma população maior, além das instâncias que compõem a amostra utilizada no estudo (BARROS; DIAS-NETO, 2011).

No restante deste trabalho, cada critério de qualidade é identificado com a letra Q e o número do critério, por exemplo, Q1 refere-se ao primeiro critério de qualidade da Tabela 3.1. Os trabalhos que foram selecionados depois de se aplicar os critérios de inclusão e exclusão descritos em nosso protocolo podem ser vistos no Apêndice A. Como pode ser visto, os estudos selecionados são classificados por ano de publicação e o título do estudo.

3.2 RESULTADOS DA REVISÃO SISTEMÁTICA E MAPEAMENTO DA LITERATURA

Esta seção apresenta os resultados produzidos pela realização da revisão sistemática e mapeamento da literatura, de acordo com o protocolo descrito na seção anterior.

3.2.1 Resultados do Mapeamento da Literatura

Nesta seção, as respostas às questões levantadas para o mapeamento são apresentadas. A Figura 3.2 apresenta os resultados relativo a distribuição das publicações através dos anos (MQ1).

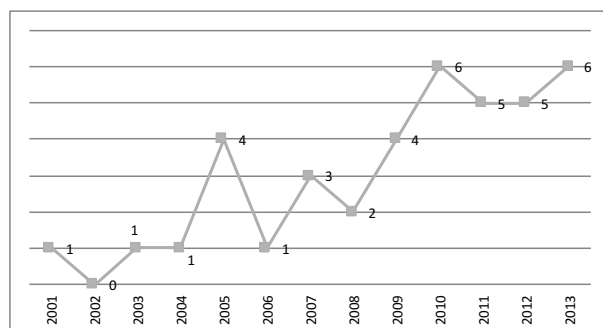


Figura 3.2: Tendências das Publicações por Ano (PITANGUEIRA; MACIEL; BARROS, 2015).

Considerando que a área é relativamente nova no campo da engenharia de Software, o baixo número de publicações nos primeiros anos e a existência de alguns altos e baixos no período total é aceitável. É importante notar o crescimento do número de estudos de 2009 que pode estar relacionado com o momento em que foi criado o Simpósio de Engenharia de Software baseada em Buscas (SSBSE).

Sobre os autores que frequentemente publicam nessa área (MQ2), a Figura 3.3 mostra o cenário. É importante salientar que, às vezes, um mesmo artigo é publicado por mais de um autor. Por exemplo, Harman e Zhang publicaram vários artigos em conjunto. Outro importante dado relevante: o estudo intitulado “*The Next Release Problem*” (BAGNALL; RAYWARD-SMITH; WHITTLEY, 2001) foi o estudo mais citado pelos artigos analisados nesta revisão sistemática e no mapeamento.

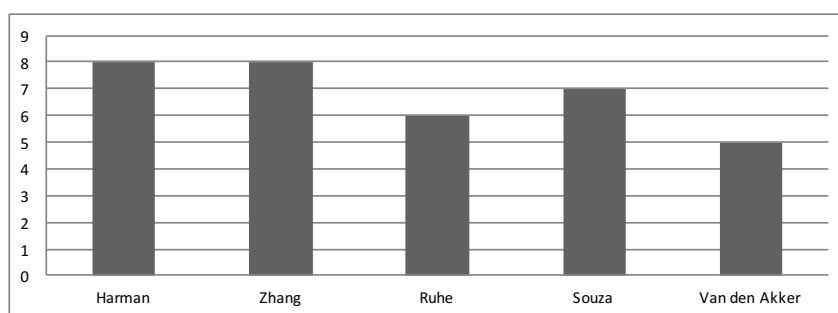


Figura 3.3: Pesquisadores mais Relevantes (PITANGUEIRA; MACIEL; BARROS, 2015).

A Figura 3.4 mostra os resultados para pergunta MQ3, que está relacionada com os principais fóruns de publicações na área de seleção de requisitos com uso de SBSE. O SSBSE foi o simpósio mais utilizado. Entre periódicos, o JISWT foi o veículo mais utilizado para publicação, seguido do periódico *Requirements Engineering*.

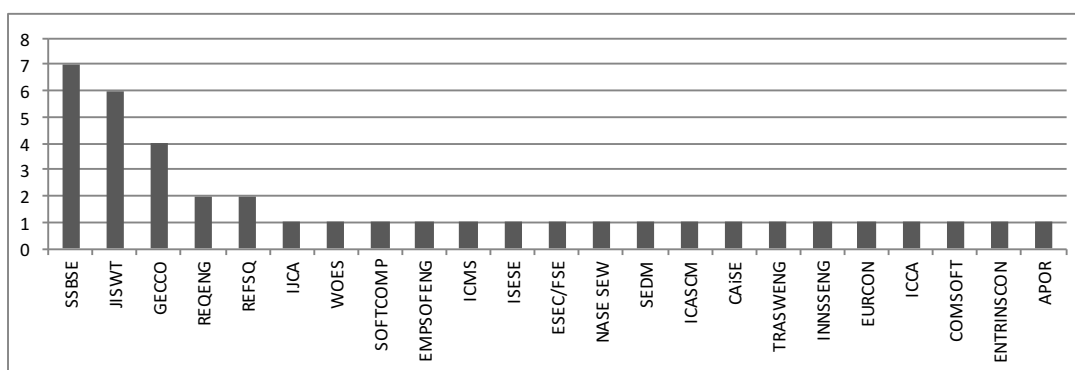


Figura 3.4: Fóruns de Publicações (PITANGUEIRA; MACIEL; BARROS, 2015).

A Figura 3.5 apresenta um resumo, destacando a distribuição temporal dos trabalhos selecionados (de 2001 a 2013), separados por tipo de publicação (*workshop*, *conference*, *journal* e *Symposium*), e especifica o número de publicações por ano e tipo de evento.

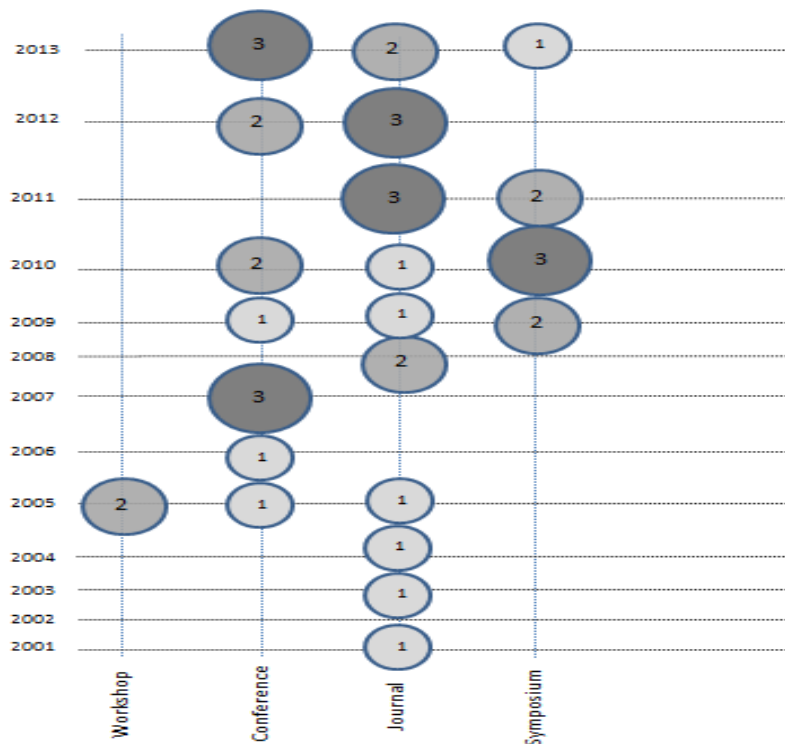


Figura 3.5: Distribuição anual por tipo de evento (PITANGUEIRA; MACIEL; BARROS, 2015).

Com esses resultados, é possível que um pesquisador ou um engenheiro de Software identifique os fóruns específicos de publicações e encontre os artigos desejados, sabendo quais autores que têm uma participação importante na seleção de software usando SBSE. Este dados subsidiam os resultados apresentados na revisão sistemática, onde são mostrados cenários específicos e a avaliação qualitativa dos estudos selecionados.

3.2.2 Resultados da Revisão Sistemática

Nesta seção, são apresentados os resultados da revisão sistemática incluindo resultados quantitativos e qualitativos. A Figura 3.6 mostra os resultados em relação ao tipo de formulação do problema abordado em seleção de requisitos de software (RQ1), destacando três tipos diferentes: estudos que usaram apenas um único objetivo (*Single objective*), estudos que usaram apenas múltiplos objetivos (*Multiple Objective*) e estudos que realizaram experiências usando ambas abordagens (*Single and Multiple Objective*).

Os resultados mostram que a formulação mono-objetivo é a formulação de problema mais utilizada na seleção de requisito de software. No entanto, desde que o trabalho intitulado *The Multi-Objective Next Release Problem* (ZHANG; HARMAN; MANSOURI, 2007) foi publicado, o número de estudos usando as formulações multi-objetivo aumentou consideravelmente. Autores geralmente concordam que esta formulação é mais eficaz e mais próxima da engenharia de software do mundo real que a formulação com um

objetivo (HARMAN et al., 2012; DURILLO et al., 2009; CAI; WEI, 2013; LI; HARMAN; LETIER, 2014).

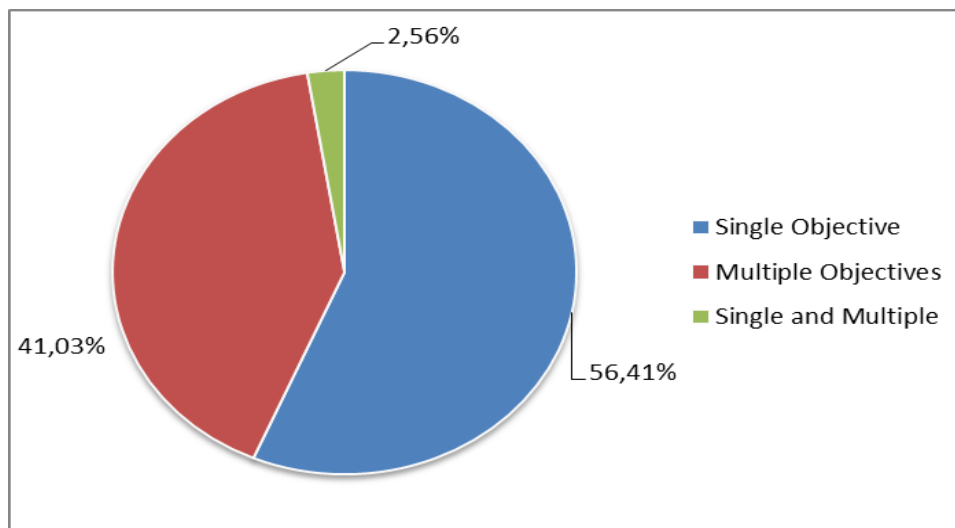


Figura 3.6: Distribuição por tipo de formulação do problema (PITANGUEIRA; MACIEL; BARROS, 2015).

A principal vantagem do MONRP é a habilidade de combinar múltiplos objetivos conflitantes, tais como maximizar a satisfação dos clientes e minimizar o esforço total envolvido no desenvolvimento dos requisitos selecionados. Além disso e diferente da formulação mono-objetivo, uma possível solução para um MONRP não é um único ponto, mas um conjunto de soluções que podem ser analisados, mostrando diferentes possibilidades e cenários para o tomador de decisões. Recentemente, algumas aplicações interessantes de MONRP têm sido utilizadas, tais como, o uso para determinação do tempo para o lançamento de uma versão do software, consideração da maximização da confiabilidade e minimização de custos (LI; XIE; NG, 2012), e a aplicação com base na satisfação dos *stakeholders*, valor e risco na gestão de negócios (BRASIL et al., 2012).

No contexto da formulação do problema, a maioria dos estudos selecionados não considerou o risco e as incertezas relacionadas ao custo e importância dos requisitos para resolver o NRP através de técnicas de otimização. Algumas abordagens que levam em consideração tais incertezas, como parte do processo de otimização, basearam-se no uso de otimização robusta para resolver esse problema. Esse é um assunto do corpo de conhecimento da pesquisa operacional que identifica e quantifica as incertezas relacionadas com o problema e constrói um modelo que busca soluções viáveis para cada realização das incertezas (BEYER; SENDHOFF, 2007).

No conjunto de dados selecionado, foram encontrados dois trabalhos relacionados à otimização robusta com esta abordagem: uma sobre um modelo robusto baseado em cenário para o NRP (PAIXAO; SOUZA, 2013a) e o outro sobre uma abordagem robusta recuperável para o NRP (PAIXAO; SOUZA, 2013b). O primeiro propõe uma nova formulação para o NRP levando em consideração as incertezas presentes nas variáveis de entrada. Neste caso, as incertezas relacionadas com a importância dos requisitos de soft-

ware foram modeladas como variáveis discretas, enquanto as incertezas relacionadas com exigência de custo foram modeladas como variáveis contínuas. O segundo estudo é uma extensão do primeiro e inclui uma solução recuperável, ou seja, para um conjunto limitado de incertezas, algumas soluções podem ser feitas viáveis ou recuperáveis, aplicando-se o esforço limitado (AHUJA; MOHRING; ZAROLIAGIS, 2009).

Em relação as restrições usadas nas formulações do problema, a maioria dos estudos usa orçamento como restrição relacionada ao custo total, enquanto alguns estudos incluem dependências entre os requisitos, limitações de tempo e fatores relacionados ao recurso (HARMAN et al., 2012; SAGRADO; Del Aguila; ORELLANA, 2010).

Em relação a hipótese (H1) - se a quantidade de estudos usando o problema com formulação multi-objetivo é uma tendência recente e sua utilização tem aumentado, quando comparado com o número de estudos realizados em um único objetivo, parece ser uma tendência recente. No entanto, observando e analisando o conjunto total de estudos em nossa revisão sistemática, normalmente, quando uma nova abordagem é pesquisada, a primeira formulação usada é a mono-objetiva e, após algumas conclusões e experimentações, é usada a formulação multi-objetivo, visando obter resultados mais precisos, desde que, no mundo real, diversos objetivos concorrentes aparecem em problemas de Engenharia de software (ZHANG; HARMAN; MANSOURI, 2007; COELLO; LAMONT; VELDHUIZEN, 2007; COELLO, 2000).

Para uma melhor compreensão do problema abordado, a Figura 3.7 mostra os resultados. Na revisão sistemática, decidiu-se separar as categorias em três, com o intuito de seguir o problema abordado e especificado pelos autores em seus trabalhos. São elas: 1) *Release*, que engloba NRP, MONRP e RP (*Release Planning*); 2) RIM (*Requirements Interaction Management*), que diz respeito ao gerenciamento e análise das interdependências dos requisitos ao longo do projeto de software (ZHANG; HARMAN, 2010) e 3) priorização (*prioritization*) que é usado para indicar uma ordem para a implementação dos requisitos (LIMA et al., 2011). De acordo com Zhang et al. (ZHANG; FINKELSTEIN; HARMAN, 2008; ZHANG; HARMAN, 2010) os conceitos e definições de NRP, RP, RIM e priorização são entrelaçados e intercambiáveis entre si, dependendo do ponto de vista do pesquisador.

Na Figura 3.7, o problema mais frequentemente abordado está relacionado a *Release* (NRP, MONRP e RP), seguido de priorização de requisitos e, finalmente, RIM. Praticamente, os mais utilizados são *Next Release Problem* e *Multi-Objective Next Release Problem*, sendo que *Release Planning* tem uma participação significativa no problema de seleção de requisitos de software.

Relativo à questão de pesquisa 2 (RQ2), sobre que técnicas são usadas para seleção de requisitos de software com foco em SBSE, o resultado pode ser visto na Figura 3.8. Elas podem ser divididas em quatro grupos:

- Meta-heurística (*Meta-heuristic*)
- Técnica exata (*Exact*)
- Técnica Híbrida usando mais de uma técnica (*Hybrid*)
- Desenvolvimento e aplicação de uma nova técnica (*New*)

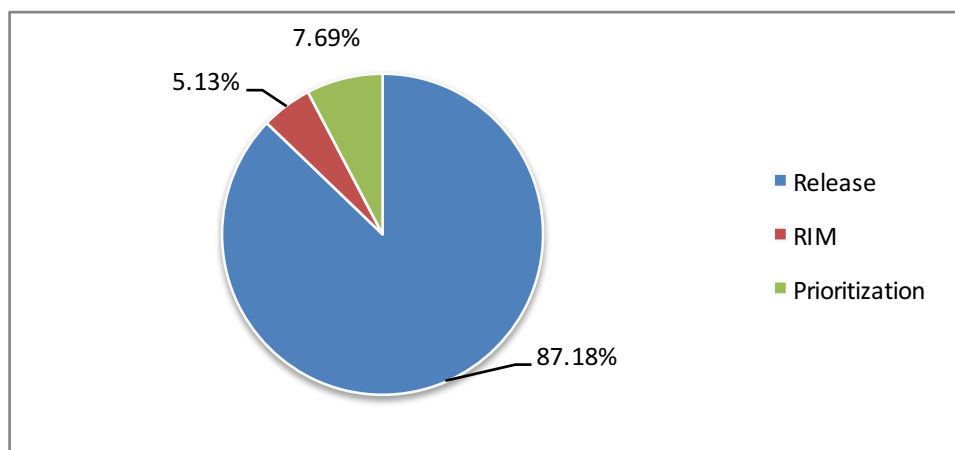


Figura 3.7: Tipo de Abordagem para Seleção de Requisitos de Software em SBSE (PITANGUEIRA; MACIEL; BARROS, 2015).

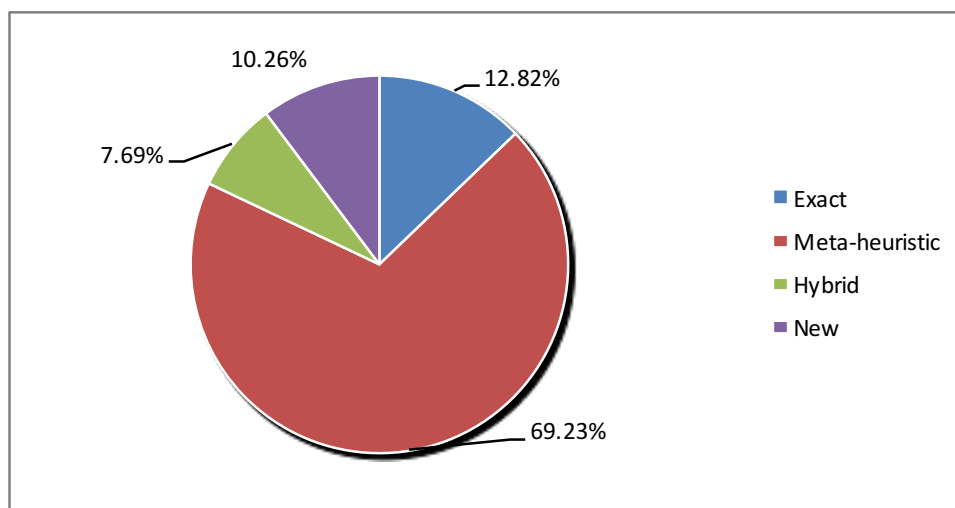


Figura 3.8: Técnicas para Seleção de Requisitos de Software em SBSE (PITANGUEIRA; MACIEL; BARROS, 2015).

Pode-se observar que técnicas baseadas em meta-heurística são mais comumente aplicadas nos experimentos visando a busca de soluções para os problemas de seleção de requisitos de software, confirmando assim que a hipótese 2 é verdadeira. De fato, tendo em conta as complexas inter-relações entre os requisitos, geralmente é bastante custoso para encontrar um conjunto de soluções que seja ótima. Portanto, o uso de meta-heurística pode ser mais apropriado quando há incerteza e dificuldade em encontrar uma frente de Pareto que mostre todas as soluções ótimas.

Em relação às técnicas híbridas (ou hibridização), poucos estudos adotaram essa abordagem, por exemplo, Jiang et al. (JIANG et al., 2010) usaram uma combinação de otimização com colônia de formigas e *Hill-Climbing*, denominando-a HACO (*Hybrid Ant Colony Optimization*). De acordo com esse autores, a técnica HACO superou a técnica

anterior em termos de qualidade da solução e tempo de execução. Outras possibilidades foram usadas, como a integração entre Programação Linear Inteira e RCPSP (Li et al., 2010) e um híbrido de decomposição e dominância com base em algoritmo evolutivo proposto por Cai e Wei (CAI; WEI, 2013), que comparou sua abordagem com NSGA-II e SPEA-2 no MONRP, obtendo melhores resultados com essa proposta híbrida. Esses métodos mostraram um potencial para resolver necessidades reais na seleção de requisitos de software. No geral, como estão a ser desenvolvidas novas técnicas, há sempre a possibilidade de combiná-las para melhorar o desempenho na tarefa de se encontrar soluções melhores.

Em relação aos outros métodos adotados, o método exato é focado normalmente na Programação Linear Inteira (AKKER et al., 2005b) ou em novos métodos desenvolvidos por alguns autores para resolver o problema, como o método de EVOLVE (RUHE; TN; GREER, 2003) e suas variações (GREER; RUHE, 2004; SALIU; RUHE, 2005b).

Finalmente, a Figura 3.9 mostra a distribuição dos resultados relativo ao tipo de técnicas de busca que são usadas para o processo de seleção de requisitos com foco em SBSE (RQ 3) e correlacionadas com a hipótese 3. Várias técnicas de pesquisa são usadas em experiências para obter boas soluções. De acordo com os dados obtidos, é evidente que o NSGA-II é a técnica mais utilizada, seguido pelo algoritmo genético (GA), Programação Linear Inteira (ILP), *Simulated Annealing* e *GREEDY*.

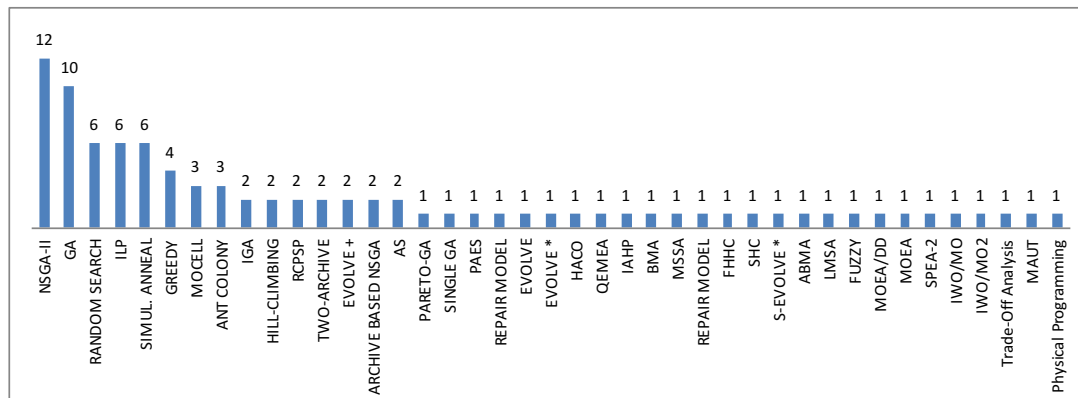


Figura 3.9: Técnicas de Busca mais utilizadas (PITANGUEIRA; MACIEL; BARROS, 2015).

Muitos autores consideram o NSGA-II a técnica mais rápida e mais eficaz para uma busca em espaços grandes e complexos (ZHANG; HARMAN; MANSOURI, 2007; ZHANG; HARMAN; LIM, 2013; VEERAPPA; LETIER, 2011; HARMAN et al., 2009). Sobre os experimentos, alguns aspectos positivos foram destacados por determinados autores sobre algumas técnicas importantes. Por exemplo, Zhang et al. (ZHANG; HARMAN; MANSOURI, 2007) apresentaram provas de que o NSGA-II é bem adequado para a formulação do problema MONRP, e que esta técnica é capaz de encontrar um conjunto de solução mais diversificado para o MONRP. Os autores observaram em seus experimentos que o NSGA-II superou outras técnicas, por exemplo, *Pareto-GA*, em encontrar a Frente de Pareto num espaço de busca muito grande.

Durillo et al (DURILLO et al., 2011) apresentaram alguns resultados interessantes, comparando algumas técnicas e mostrando suas vantagens específicas. Em seus experimentos, a análise mostrou que enquanto o NSGA-II foi a técnica que encontrou um maior número de soluções ótimas, MOCell ofereceu aos tomadores de decisão uma maior gama de soluções diferentes. Em relação ao tempo de execução, PAES foi a técnica mais rápida, embora com resultados menos precisos.

Uma observação importante está relacionada ao uso da busca aleatória (*Random Search*) em experimentos. Apesar da alta ocorrência do mesmo, como visto na Figura 3.9, esta técnica é usada para a comparação de resultados e não é considerada como um método para resolver o problema da seleção requisitos de software. Para resumir, apesar do grande uso do NSGA-II, a Figura 3.9 mostra uma grande diversidade de técnicas de pesquisa utilizadas e isso pode indicar que não há nenhum consenso nesta área em relação a melhor técnica para resolver o problema complexo da seleção de requisitos de software, confirmando a hipótese 3.

Além disso, outros resultados foram importantes para mapear a área de pesquisa estudada na revisão sistemática. As Figuras 3.10 e 3.11 a seguir mostram os resultados referentes a dados utilizados nos experimentos.

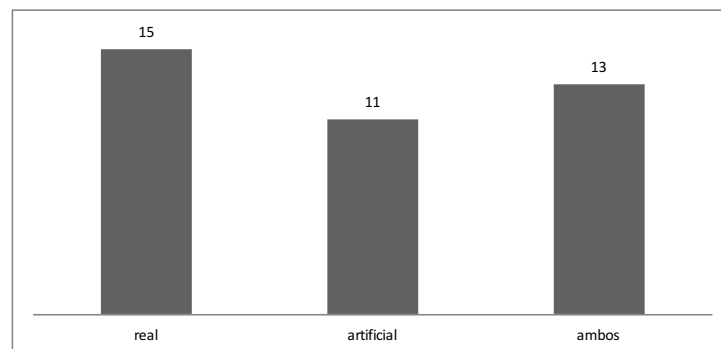


Figura 3.10: Tipos de dados usados nos experimentos (PITANGUEIRA; MACIEL; BARROS, 2015).

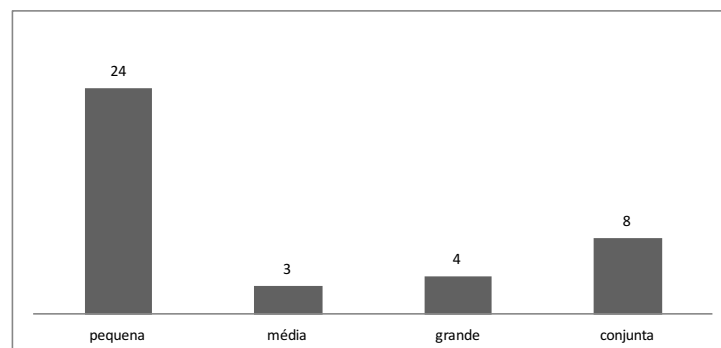


Figura 3.11: Tipos de dados usados nos experimentos (PITANGUEIRA; MACIEL; BARROS, 2015).

Por exemplo, a Figura 3.10 mostra os experimentos que usaram tipos de dados reais (normalmente obtidos na indústria de desenvolvimento de software), dados artificiais e alguns estudos que relataram tanto o uso de dados reais como dados artificiais em seus experimentos. Praticamente, 50% dos estudos utilizaram exclusivamente dados reais.

É importante ressaltar que, a organização e catalogação desses resultados gerou um trabalho extra em nossa revisão sistemática e mapeamento porque muitos estudos não explicavam, claramente, as instâncias selecionadas usadas em suas experiências, às vezes, citando apenas um banco de dados ou dados utilizados por outros autores, sem mencionar o tipo de dados de forma explícita.

Em relação a escala de dados usada nos experimentos, a Figura 3.11 mostra os quatro tipos: pequena escala, escala média, em grande escala e a quantidade de estudos que usaram uma combinação de dois ou mais tipos de instância em seus experimentos.

Para essa classificação das grandezas das instâncias de dados, foi tomado como base a classificação adotada por Zhang et al. (ZHANG; HARMAN, 2010). Neste caso, a escala é proporcional ao número de requisitos, denotando, em pequena escala (requisitos de 1-100), média (101-250 requisitos) e em grande escala (251 em diante). A maioria dos estudos usou instâncias em pequena escala ou uma combinação de instâncias de pequenas e médias em seus experimentos.

Em relação aos critérios de qualidade (ver Tabela 3.1), as figuras a seguir mostram os resultados divididos por tipo de construto, de acordo com a Tabela citada. A Figura 3.12 mostra os resultados que dizem respeito a validade geral. Nela, é possível identificar que todos os critérios obtiveram excelentes valores. Apenas o critério 5, que aborda se o estudo faz uma projeção de valor para a pesquisa e a prática, mostrou uma percentagem mais acentuada para a não-observância desse critério. De fato, cerca de 38% dos artigos analisados não identificaram as novas áreas de aplicação para a pesquisa e seu uso em diferentes populações.

A Figura 3.13 mostra os resultados obtidos em relação a validade interna. Estes dados

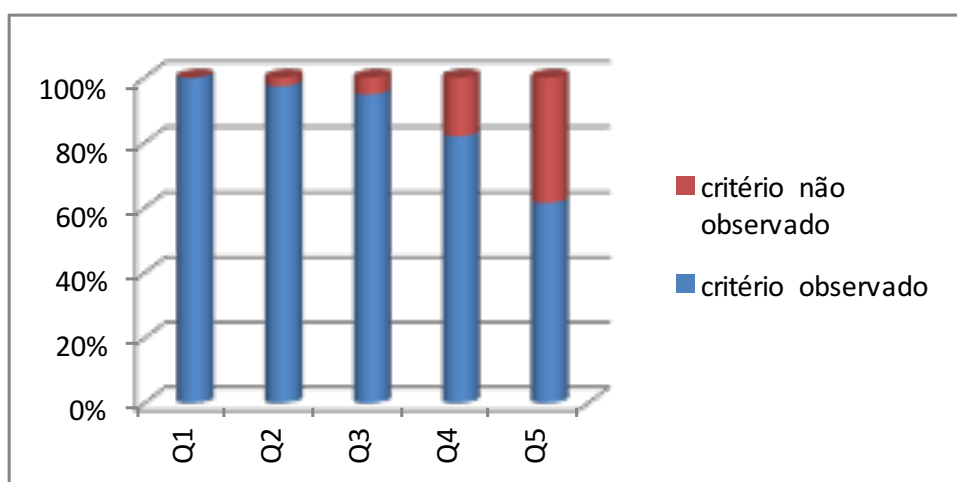


Figura 3.12: Resultados para os critérios de validade geral (PITANGUEIRA; MACIEL; BARROS, 2015).

mostram que quase metade dos estudos não fornece o código-fonte usado nos experimentos (Q6) e uma quantidade significativa de estudos não detalha o procedimento para coletar os dados utilizados no experimento (Q7). Quanto à pergunta 8, relativa à definição de instâncias de destino (aleatória ou real), a maioria dos estudos satisfaz este critério.

Validade de conclusão é provavelmente o construto que requer especial atenção devido às taxas obtidas. A Figura 3.14 mostra os dados obtidos. Sobre a Q9 (se o experimento considera variações aleatórias (diferentes execuções nos algoritmos apresentados), praticamente, 38% dos experimentos não consideram variações aleatórias. No caso da questão 10 (Q10), que avalia se a hipótese do estudo foi apresentada formalmente, mais de 71% dos estudos selecionados não observaram esse critério. No que diz respeito ao critério de qualidade 11 (Q11- uso de teste de inferência estatística), somente 13 trabalhos observaram este critério e, finalmente, sobre a Q12 (se havia uma base significativa de comparação entre as técnicas utilizadas e as soluções obtidas pelas técnicas mais conhecidas), 48.71% cumprem este critério de qualidade.

Em relação aos itens que abordam a validade de construção (Figura 3.15), bons números são mostrados para a maioria dos critérios. Em relação ao critério 13 (se o artigo discute o modelo adotado para o processo de otimização) 38 estudos dos 39 selecionados atendem positivamente a este critério, enquanto 32 estudos atendem ao critério 14 (Q14) (se o artigo discute a validade das medidas de eficácia).

Finalmente, sobre a validade externa, a Figura 3.16 mostra os resultados obtidos. Nessa figura, pode se identificar que para o critério 18, relativo à apresentação dos valores de parâmetros utilizados no estudo, 100% dos estudos atendem esse critério, no entanto, para o critério 15 (se o estudo mostra claramente a estratégia de seleção de

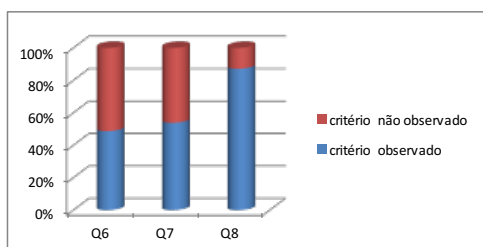


Figura 3.13: Resultados para a validade interna

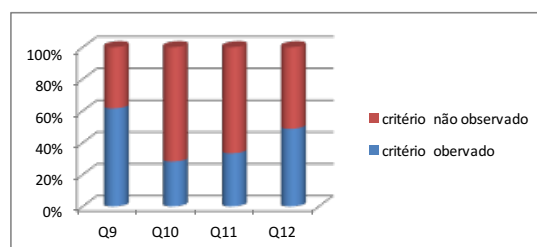


Figura 3.14: Resultados para validade de conclusão

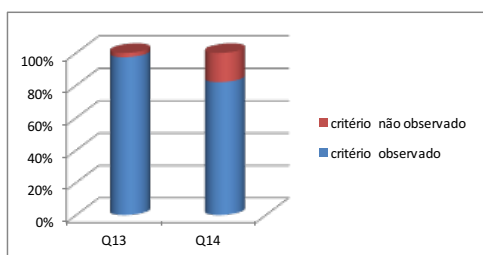


Figura 3.15: Resultados para a validade de construção

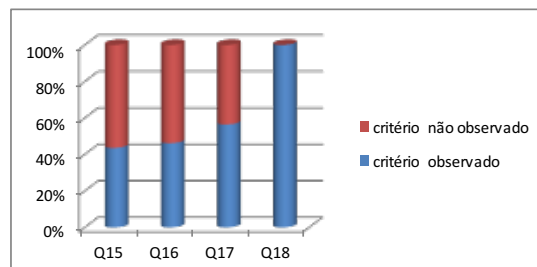


Figura 3.16: Resultados para validade externa

instâncias (dados reais ou gerado aleatoriamente) apenas 43,5% dos estudos atendem essa perspectiva e, para o critério Q16 (se o estudo aborda a variação na complexidade das instâncias usadas no experimento) 46% abordam a variação. Finalmente, sobre se o estudo utiliza instâncias do mundo real na experiência ou não (Q17), mais de 20 estudos afirmaram ter usado instâncias do mundo real.

3.3 RESUMO DOS RESULTADOS DA REVISÃO SISTEMÁTICA E MAPEAMENTO DA ÁREA

Após a apresentação dos resultados, os dados qualitativos extraídos foram analisados, mostrando tendências recentes, possibilidades, pontos fortes e fracos no campo de estudo proposto nessa revisão sistemática. Inicialmente, os resultados indicaram um número crescente de publicações a partir de 2009 (Figura 3.2). Esse dado possui relação direta com a criação do Simpósio Internacional sobre Engenharia de Software baseada em Buscas (SSBSE), onde o programa da conferência adotou múltiplas áreas, e especificamente em 2010, uma área foi dedicada exclusivamente à seleção de requisitos de software.

Em geral, os resultados para os critérios de qualidade mostram quais critérios devem ser observados para o planejamento e execução de experimentos que possam ser bem elaborados e confiáveis. Quanto aos estudos selecionados, algumas observações e percepções importantes podem ser resumidas:

1. Tendência para o uso de formulação multi-objetivo;
2. Incorporação de novas técnicas de busca, principalmente, vindas da pesquisa operacional;
3. Uso de dados reais em pequena escala;
4. Os tipos de requisitos não são categorizados (funcionais ou não funcionais);
5. Ainda poucos estudos consideram as interdependências entre os requisitos;
6. A maioria das modelagens considera apenas custo e valor;
7. Poucos estudos incluem o conceito de risco nas suas formulações.

Também, através dos resultados, avaliação e análise crítica dos estudos selecionados, foi possível identificar algumas áreas em aberto e tendências. Portanto, alguns pontos de interesse podem ser levados em consideração para novas pesquisas.

De início, o uso de dados reais é uma boa via para a experimentação e obtenção de resultados, dado a proximidade com fatos e características que pertencem ao desenvolvimento de software nas indústrias. No entanto, é sabido que a obtenção desse tipo de dados muitas vezes é extremamente complicado, devido às restrições de sigilo impostas pela indústria de software e seus clientes (ZHANG; HARMAN, 2010; VEERAPPA; LETIER, 2011; BECHIKH; SAID; GHÉDIRA, 2013).

Outro ponto importante é a consideração da interdependência entre os requisitos de software. Ainda poucos estudos aplicam as relações entre os requisitos. Os tipos

3.3 RESUMO DOS RESULTADOS DA REVISÃO SISTEMÁTICA E MAPEAMENTO DA ÁREA 45

mais comuns de relação são: precedência, relacionadas com valor, relacionadas a custo, operadores *AND*, *OR* e *exclusive OR*. Obviamente, tais características dependem de como os dados estão organizados e, nem sempre, as empresas disponibilizam essas dependências. Muitas vezes, há a necessidade de se criar tais relações entre os requisitos (AKKER et al., 2005a; ZHANG; FINKELSTEIN; HARMAN, 2008).

Uma outra observação importante diz respeito ao uso de técnicas exatas para a busca de soluções, tais como, programação inteira e *constraints solvers*. Certamente, a busca exata exige tempo de processamento e hardware potente. No entanto, há a possibilidade de instâncias de médio e pequeno porte serem abordadas com o uso da busca exata, antes de se aplicar diretamente um método probabilístico (REGNELL; KUCHCINSKI, 2011; TONELLA; SUSI; PALMA, 2010, 2013).

Outra possibilidade de direcionamento de estudo na área, diz respeito à inclusão das preferências dos usuários para obtenção de soluções, seja ela feita *a priori*, *a posteriori* ou *interativa*. A maioria dos estudos não considerou essas possibilidades explicitamente. Além disso, poucos estudos levam em consideração a participação interativa dos usuários (TONELLA; SUSI; PALMA, 2010, 2013).

Em termos de novas perspectivas, alguns pontos importantes puderam ser identificados:

1. Realização de experimentos em SBSE que categorizem e diferenciem os requisitos funcionais e não-funcionais, dado que existem diferenças entre eles, assim, novas experiências podem ser executadas e os resultados obtidos podem corresponder à realidade da indústria de software. Atualmente, não existem estudos em NRP/MONRP que realizem essa diferenciação e as suas consequências.
2. Adição das preferências e julgamento do usuário na modelagem do problema de seleção de requisitos de software, seja para se determinar Regiões de Interesse ou direcionar a busca para soluções específicas na Frente de Pareto. Isso pode levar a melhores resultados e experiências mais próximas dos desejos dos *stakeholders*.
3. Inclusão de novas restrições e objetivos que possam fazer com que as formulações sejam mais próximas da realidade da indústria de software, como por exemplo, incertezas inerentes às soluções, riscos relacionados ao orçamento, riscos relacionados à satisfação do usuário, riscos relacionados com os prazos do projeto e fatores de consumo de recursos e restrições de qualidade que podem ter um impacto significativo no desenvolvimento de software.

Com os resultados quantitativos e qualitativos e a análise de temas ou pontos de interesse ainda em aberto ou pouco explorados, foi possível situar o contexto e abordagem para esse trabalho de tese, principalmente, no que diz respeito ao uso das técnicas de busca e procedimentos para a experimentação para o trabalho de tese.

Assim, a tendência para a formulação multi-objetivo foi adotada na tese, além de considerar, uma nova formulação para o MONRP levando em consideração o risco associado à insatisfação dos *stakeholders* quanto a percepção do valor dos requisitos em cada solução, algo ainda inexplorado. Adicionalmente, foram usadas técnicas de busca exata

baseada em *SMT Solvers*, sendo que uma delas não foi aplicada em nenhum estudo selecionado. Em sequência, o NSGA-II, meta-heurística comprovadamente bastante utilizada no problema de seleção de requisitos de software foi usado e seus resultados comparados aos *SMT Solvers*.

3.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou uma revisão sistemática e um mapeamento da literatura sobre a seleção de requisitos de software usando técnicas baseadas em buscas com o intuito de categorizar, identificar e classificar os estudos selecionados, mostrando um resumo visual e quantitativo dos trabalhos previamente realizados na área.

O próximo capítulo mostra e discute os conceitos de risco, apresentando algumas categorizações de risco de software e formas de se analisar riscos em projetos de software.

RISCOS EM PROJETOS DE SOFTWARE

Um risco é qualquer potencial situação ou evento que possa afetar negativamente a capacidade do projeto em atingir os seus critérios de sucesso expressos e esperados pelos *stakeholders*. Em outras palavras, um risco é um fator ou elemento que envolve uma exposição a perdas ou danos (HOODAT; RASHIDI, 2009).

De acordo com Lamsweerde (LAMSWEERDE, 2009), risco é um fator de incerteza que tem uma probabilidade de ocorrência e, caso se torne real, pode resultar em perda de satisfação a um ou mais objetivos da empresa, tendo um impacto negativo sobre estes objetivos. Assim, o risco pode ser descrito como o valor esperado dos danos causados ao projeto por uma combinação de condições incertas.

O SWEBOK (FAIRLEY; BOURQUE, 2014) descreve que uma situação de risco é caracterizada pela probabilidade de um evento ocorrer e resultar em um impacto negativo sobre um projeto, tendo como consequência resultados inconvenientes, sofrimento ou perda perceptível para o projeto. De acordo com Meli (MELI, 1998, 1999), para que o risco se materialize, devem ser atendidas as seguintes condições:

- (a) uma ou mais situações de incerteza devem estar presentes no ambiente da empresa;
- (b) deve haver um evento ou combinação de eventos com uma probabilidade não nula de ocorrência;
- (c) deve haver uma relação de causa e efeito desencadeada por estes eventos e capaz de determinar um impacto e,
- (d) esta situação deve ser considerada negativa com respeito às expectativas e escala de valores dos *stakeholders*.

4.1 CATEGORIZAÇÃO DOS RISCOS DE SOFTWARE

A categorização dos riscos é uma forma de classificação para melhor compreensão dos riscos inerentes ao projeto de software. Na literatura sobre riscos de software existem diversos estudos que visam classificar os riscos que afetam projetos de software (AS-NAR; GIORGINI; MYLOPOULOS, 2011; AMBER; SHAWOO; BEGUM, 2012; ISLAM; HOUMB, 2010) e, de acordo com eles, há tantos riscos que é muito difícil (ou mesmo impossível) identificar todos eles com antecedência. Em geral, as diversas classificações, possuem similaridades, no entanto estão sujeitas a variações de contexto para contexto (LAMSWEERDE, 2009; JORGENSEN, 2010). Está fora do escopo dessa Tese fazer uma categorização dos risco de software, no entanto, algumas categorias são úteis para situar o conceito de risco trabalhado e adotado no contexto da Tese, principalmente, no que diz respeito aos riscos relacionados com os requisitos de software. Algumas categorias importantes são mostradas a seguir.

Para Yavari et al. (YAVARI, 2013), os riscos de software podem ser categorizados de forma mais ampla, como fatores de risco relacionados aos usuários, requisitos, a complexidade do projeto, planejamento / controle, equipe e ambiente organizacional. Quanto ao riscos de requisitos de software, esse autor identifica os seguintes: a) mudança contínua dos requisitos ao longo do projeto; b) requisitos de software não identificados adequadamente, e c) requisitos de software incorretos.

Hoodat e Rashidi (HOODAT; RASHIDI, 2009), agruparam e classificaram os riscos de forma mais específica como sendo, riscos de requisitos de software, riscos de custo do software, riscos relacionados a programação, riscos de qualidade de software e riscos relacionados ao negócio. Para esses autores, os riscos relacionados aos requisitos de software são os seguintes:

- (a) Má definição dos requisitos de software;
- (b) Mudança de requisitos de software;
- (c) Requisitos impossíveis de serem implementados;
- (d) Má estimativa dos requisitos (custo, importância, tempo de implementação);
- (e) Ambiguidade dos requisitos;
- (f) **Discordância entre os membros do projeto sobre a importância dos requisitos;**
- (g) Dificuldade em implementar requisitos complexos.

Rana et al. (RANA et al., 2014) propõem uma categorização dos riscos para cada versão do software, sendo estes, Planejamento das versões do software, projeto e construção de cada versão, aceitação da versão, treinamento e distribuição do software. Para esses autores, o que chama mais a atenção é o risco relacionado a atividade de planejamento da versão, em outras palavras, esse risco está associado a mudanças e *diferentes entendimentos dos stakeholders sobre quais requisitos selecionar em cada versão*. Nesse

caso, uma probabilidade de ocorrência de 80% desse risco tornar-se realidade, criaria um efeito em cascata em todas as próximas versões do software tendo um impacto negativo no processo de desenvolvimento do software.

4.2 ANÁLISE DE RISCOS DE SOFTWARE

A análise de risco é frequentemente utilizada no desenvolvimento de software para identificar eventos ou situações que podem ter um impacto negativo sobre um projeto de software (FEATHER; CORNFORD, 2003; ASNAR; GIORGINI; MYLOPOULOS, 2011).

Os riscos podem ser avaliados de forma qualitativa ou quantitativa. Por exemplo, sob o aspecto qualitativo, um risco pode ser avaliado em uma escala ordinal de ‘muito improvável’ a ‘muito provável’ (LAMSWEERDE, 2009). De forma similar, Wen-Ming e Huang (HAN; HUANG, 2007) analisam os riscos criando cinco níveis: 1) remoto, 2) improvável, 3) provável, 4) altamente provável e 5) próximo da certeza. Keil et al. (KEIL et al., 1998) classificam os riscos através da importância relativa que deve ser dada ao risco percebido pela empresa de software, sendo elas, em um intervalo de ‘menos importante’ até ‘muito importante’. Alguns autores avaliam o risco através do possível impacto no projeto de software, como por exemplo, de ‘insignificante’ a ‘catastrófico’, de acordo com a gravidade das suas consequências (LAMSWEERDE, 2009; PRESSMAN, 2001).

No que diz respeito ao aspecto quantitativo, as formas mais utilizadas são feitas através do uso de valores discretos de probabilidade (ex: 0, 0.1, ..., 1.0) (LAMSWEERDE, 2009), intervalos de probabilidade (ex: [0-0.3], [0.3-0.5], ..., [0.7-1.0]) (POHL, 2010), usando escalar ordinal de 1 até 10 (SOMMERVILLE, 2004) ou através de algum formalismo matemático, tal como, o cálculo de variância ou desvio-padrão com pesos (MOORES; CHAMPION, 1996; FEATHER; CORNFORD, 2003; RONG et al., 2009).

Por exemplo, Lamsweerde (LAMSWEERDE, 2009) calcula o risco através da exposição (E) ao risco r, da seguinte forma:

$$E(r) = \sum_c L(c) * S(c) \quad (4.1)$$

onde L(c) e S(c) são a probabilidade e severidade de consequência (c), respectivamente. Dessa forma, se pode comparar as exposições de vários riscos identificados para se considerar e dar atenção aos riscos com maior valor de exposição.

De uma forma similar, Bannerman (BANNERMAN, 2008) calcula o risco através da equação $R = P \times I$, onde R é o fator de risco, P é a probabilidade de um evento indesejável ocorrer e I é a magnitude de uma possível perda, se esse evento de fato ocorrer. Nesse caso, o risco é usualmente medido em dólares ou tempo em projetos comerciais.

Pressman (PRESSMAN, 2001) quantifica o risco também usando a exposição ao risco (RE), porém, usando a probabilidade de ocorrência de um risco (P) e o custo (C) para o projeto, caso o risco ocorra, obtendo a equação $RE = P \times C$.

Apesar dessas análise de riscos em projetos de software serem bastante utilizadas, elas não levam em consideração o risco quando há a insatisfação entre os *stakeholders* com o conjunto de requisitos que podem ser selecionados para a versão do software.

Esta Tese visa abordar esse problema através de uma nova abordagem para o cálculo

do risco relativo aos requisitos de software. Essa abordagem visa reduzir o risco associado com a insatisfação dos *stakeholders*, fator crítico para todo o planejamento do software, como exposto anteriormente. A seção 5 mostra em detalhes a proposta dessa abordagem.

4.3 TRABALHOS RELACIONADOS - ABORDAGENS PARA TRATAMENTO DE RISCOS EM SELEÇÃO DE REQUISITOS

Risco tem sido uma preocupação constante no desenvolvimento de software, uma vez que, muitos dos problemas enfrentados pelos desenvolvedores de software tem relação direta com a não consideração de alguns riscos existentes ao longo do processo de desenvolvimento do software, tais como, não envolvimento do usuário, expectativas do produto final de software não realistas, entre outros. (FAIRLEY; BOURQUE, 2014)(GROUP, 2014).

Desde que alguns problemas da engenharia de software, como por exemplo, seleção de requisitos, teste e arquitetura de software tem sido abordados e reformulados como problemas de busca (CLARKE et al., 2003), há um interesse crescente em se estudar o risco como objetivos relevantes do processo de busca (LETIER; STEFAN; BARR, 2014).

Conforme Ruhe e Greer (RUHE; TN; GREER, 2003), a inclusão do risco no processo de seleção de requisitos de software deve ser sempre levada em consideração. Nos projetos em que o risco é abordado a cada incremento, maior a possibilidade de se evitar problemas no ciclo de desenvolvimento do software, tais como, estouro do orçamento e versões de software que não estão de acordo com o especificado pelos *stakeholders*. Asnar, Giorgini e Mylopoulos (ASNAR; GIORGINI; MYLOPOULOS, 2011) acrescentam que, caso os riscos inerentes ao projeto de software não sejam abordados no processo de engenharia de requisitos, a chance do produto de software não atender às expectativas e demandas dos *stakeholders* é bastante elevada.

Considerando essa perspectiva sobre riscos, alguns trabalhos foram desenvolvidos com a inclusão do risco como objetivo a ser minimizado. Por exemplo, Ruhe e Greer (RUHE; TN; GREER, 2003) desenvolveram um estudo quantitativo para o planejamento de versões de software sob risco e limitações de recursos. Para os autores, cada risco pode ser um evento que potencialmente afeta o cronograma de um projeto de software com consequências negativas no resultado final. Nesse trabalho, o fator de risco é estimado para cada requisito e um valor máximo de referência foi calculado para cada versão do software. Uma ferramenta foi desenvolvida (EVOLVE+) para dar suporte ao processo de minimização do risco em cada incremento. A abordagem utilizada para o risco leva em consideração um determinado requisito i (r_i) e uma versão (ou incremento) do software (R^k). Então, para cada par (r_i, R^k) há uma função de escala para o risco: $(r_i, R^k) \rightarrow [0,1)$, em que '0' significa ausência de risco e '1' representa o maior risco. Segundo Ruhe e Greer (RUHE; TN; GREER, 2003), a ideia principal dessa abordagem visa um balanceamento do conjunto de requisitos para que não houvesse concentração de requisitos que possuíssem altos valores de risco a serem desenvolvidos em um mesmo incremento.

Os resultados encontrados mostram que, na abordagem desenvolvida por esses autores, é possível obter soluções factíveis, que proporcionam um equilíbrio entre os interesses conflitantes dos *stakeholders*, pondo em evidência o risco estimado para cada versão de software.

Similar ao trabalho anterior, no que diz respeito ao gerenciamento de risco ao longo de cada versão ou incremento do desenvolvimento de software, Colares et al. (COLARES et al., 2009) projetaram uma formulação multi-objetivo para o planejamento de versões do software levando em consideração: 1) a maximização da satisfação dos *stakeholders* e 2) a minimização dos riscos de projeto, enquanto respeitavam as restrições relativas as disponibilidade de recursos e as dependências entre os requisitos. Basicamente, a formulação do risco é a seguinte forma:

$$\text{Min} \sum_{r \in R}^n \text{risk}_r \cdot x_i \quad (4.2)$$

onde risk_r representa o risco associado a um requisito e x_i a variável booleana (1 ou 0) associada a inclusão ou não de um requisito. Os autores adotaram como medida para o risco de cada requisito, uma escala *Likert* (LIKERT, 1932) variando de 0 (menor valor) a 5 (maior valor).

Esse trabalho utiliza como técnicas de busca o NSGA-II e a busca aleatória. Adicionalmente, cinco profissionais da área foram convidados para projetar as versões do software, usando os mesmos dados. Os resultados encontrados mostram que o NSGA-II possui melhores resultado do que a busca aleatória e os profissionais usados no processo de otimização.

Uma nova formulação proposta por Brasil et al.(BRASIL et al., 2012) inclui a satisfação dos *stakeholders*, a priorização dos requisitos, o valor de negócio e o gerenciamento do risco ao longo do desenvolvimento do software, enquanto se respeita a limitação de recursos e as dependências entre os requisitos. O risco utilizado nesse trabalho está associado a possíveis problemas que possam existir no projeto de software, afetando negativamente o gerenciamento desse projeto, como por exemplo, o risco de erros na atividade de priorização dos requisitos que podem influenciar a ordem dos requisitos a serem implementados e, conseqüentemente, afetar as versões de software a serem desenvolvidas.

Portanto, os requisitos que possuem alto valor de risco devem ser implementados o mais cedo possível, isto é, nas primeiras versões do software. Isto porque, conforme esses autores, requisitos com alto valor de risco associado, são mais propensos a afetar potencialmente o produto final do software, então, quanto mais cedo seja essa implementação, menos complicada será a gestão de problemas advindos desses tipos de requisitos. Conforme os autores a equação 4.3 expressa o gerenciamento do risco no projeto como um todo, ou seja, em cada versão do software.

$$\text{Min} f_{\text{risk}}(x^{\text{prio}}) = \sum_{i=1}^N \text{risk}_i \cdot x_i^{\text{prio}} \quad (4.3)$$

A variável x_i^{prio} indica a posição do requisito r_i na ordem de prioridade previamente estabelecida pelos *stakeholders*, podendo assumir valores de 0 a N para cada $i = 1, 2, \dots, N$. A variável risk_i representa o risco associado a cada requisito i . Para obter os

resultados, os autores usaram duas meta-heurísticas conhecidas, NSGA-II e MOCell em bases de dados artificiais, sendo que MOCell apresentou os melhores resultados. Os resultados com o uso dessas técnicas para a formulação proposta, mostra que é possível encontrar soluções viáveis que minimizem o risco de conflitos no processo de priorização de requisitos.

Li et al. (LI; HARMAN; LETIER, 2014) desenvolveram uma formulação robusta para lidar com o risco durante o processo de otimização. Para esses autores, o risco tem impacto significativo no desenvolvimento de software porque pode gerar atraso na entrega da versão final ou um produto incompatível com os desejos dos usuários.

A formulação desse trabalho levou em consideração três objetivos: custo, receita e risco. Além disso, no caso do custo, os autores entendiam a incerteza como a probabilidade de que o custo real exceda um limite prévio de custo estipulado. Assim, o risco de uma possível solução pode ser medido pela probabilidade que o custo dessa solução exceda um limite total de custo previamente estabelecidos pelos tomadores de decisão.

O cálculo do risco foi simulado por meio de variáveis estocásticas através de distribuição de probabilidades. Portanto, a fim de reduzir o risco em extrapolar o orçamento, a abordagem adotada minimiza a probabilidade de que o custo real exceda o orçamento geral da companhia de software, conforme a equação a seguir:

$$Risk(\vec{x}) = Pro(custo_atual(\vec{x}) > \Theta \cdot Custo_Esperado(\vec{x})) \quad (4.4)$$

onde Θ é a percentagem designada pelo tomador de decisão (ex., $\Theta = 150\%$), e *Pro* significa a Probabilidade. Por consequência, a função objetivo correlacionada deve minimizar o risco, de acordo com a seguinte equação:

$$Minimize(\vec{x}) = Risk(\vec{x}) \quad (4.5)$$

Os resultados apresentados por Li et al. (LI; HARMAN; LETIER, 2014) mostram que a formulação para o risco auxilia os tomadores de decisão no processo de exploração de soluções com baixo risco, evitando assim uma possível extrapolação do orçamento. Além disso, os autores perceberam em seus achados que enquanto o custo está intimamente correlacionado com a inclusão de um requisito na Frente de Pareto, a receita não está.

A Tabela 4.1 mostra um resumo dos trabalhos relacionados identificando o tipo de risco abordado, as técnicas aplicadas para resolver o problema e as contribuições.

Tabela 4.1: Abordagens para Tratamento de Riscos em Seleção de Requisitos

Trabalho	Tipo de risco	Técnicas de busca	Contribuições
(RUHE; TN; GREER, 2003)	Risco no planejamento de versões do software	EVOLVE+	Balanceamento do risco ao longo das versões do software
(COLARES et al., 2009)	Risco no planejamento de versões do software	NSGA-II, Random Search	Controle e gerenciamento do risco no desenvolvimento de um software
(BRASIL et al., 2012)	Risco no planejamento de versões do software	NSGA-II, MOCcell	Formulação combinando o risco e a priorização de requisitos
(LI; HARMAN; LE-TIER, 2014)	Risco de estouro do orçamento	Método de Monte Carlo	Formulação para risco de requisitos em orçamento de projeto

Como pode ser visto na Tabela 4.1, a maioria dos trabalhos adota o risco no planejamento das versões do software visando o gerenciamento dos riscos ao longo do ciclo de desenvolvimento de um projeto de software. Através das formulações propostas por cada trabalho, foi possível encontrar soluções viáveis que minimizassem os efeitos dos riscos para um projeto de software.

Esses trabalhos são estudos iniciais importantes para a inclusão do risco no processo de seleção de requisitos de software, abordando aspectos relevantes do risco em projetos de software. No entanto, como há diversas categorizações e tipos de risco de software, alguns tipos de risco não foram ainda trabalhados ou analisados profundamente, sendo esse o caso do risco da insatisfação dos *stakeholders* na seleção de requisitos de software.

A principal diferença deste trabalho de Tese para os trabalhos acima citados é uma nova abordagem para o risco, nesse caso, o risco de insatisfação dos *stakeholders* no processo de seleção de requisitos para cada incremento do software. Para tanto, uma nova formulação multi-objetivo para o problema de seleção de requisitos para a próxima versão do software foi desenvolvida, com o objetivo de minimizar o risco da insatisfação dos *stakeholders* (RA-MONRP).

O risco de insatisfação dos *stakeholders* é calculado com base nas variações de atribuições de valor/importância para os requisitos de software. A abordagem desse trabalho de Tese considera a variância como foco principal para expressar esse risco permitindo assim, acomodar os diversos pontos de vistas e opiniões distintas dos *stakeholders* em relação a importância dos requisitos e, por consequência, das soluções que contêm subconjuntos de requisitos. Em relação às técnicas adotadas para o processo de busca, diferentemente dos trabalhos relacionados que usaram métodos probabilísticos, uma técnica determinística baseado na *Satisfiability Modulo Theory* (SMT) bastante utilizada na área de pesquisa operacional (HILLIER; LIEBERMAN, 2004) foi utilizada para toda a abordagem do RA-MONRP.

4.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o conceito de risco, as categorias de riscos de software, algumas formas de se analisar e abordar um risco no ciclo de desenvolvimento de software e como

alguns trabalhos existentes integram a análise do risco durante o processo de seleção de requisitos para o desenvolvimento dos incrementos do software.

O próximo capítulo apresenta a abordagem proposta para o risco de insatisfação dos *stakeholders* durante o processo de seleção de requisitos de software (RA-MONRP), mostrando a formulação para o RA-MONRP e as técnicas de busca utilizadas para esse problema.

ABORDAGEM PROPOSTA

O problema da seleção de requisitos de software envolve um processo de tomada de decisão não estruturado ou vagamente estruturado caracterizado por uma diversidade de fatores, como a grande quantidade de requisitos que devem ser avaliados, escassez de recursos, diferenças de percepção entre os múltiplos *stakeholders* que participam no processo de decisão, entre outros (RUHE; TN; GREER, 2003). Na presença de vários *stakeholders*, a seleção de um subconjunto de requisitos é uma tarefa desafiadora, porque os *stakeholders* podem ter diferentes aspirações, desejos e opiniões (ORTEGA et al., 2013). Conseqüentemente, suas preferências podem ser conflitantes e suas avaliações de valor podem variar muito por causa dos seus diferentes pontos de vista (VEERAPPA; LETIER, 2011; ZHANG; HARMAN; MANSOURI, 2007).

Neste contexto, a decisão de quais requisitos serão implementados na próxima versão de um projeto de software torna-se muito complexa (HARMAN et al., 2012) e propensa à insatisfação dos *stakeholders* (SAGRADO et al., 2010; LETIER; STEFAN; BARR, 2014; FAIRLEY; BOURQUE, 2014; WICKRAMASINGHE W; LI, 2008). A incerteza associada com tal processo de tomada de decisão pode ser atenuada se os riscos relacionados ao atendimento das aspirações dos *stakeholders* forem explicitamente levados em consideração e minimizados. Desta forma, pode-se evitar que um conjunto de soluções que não leve em consideração aspectos técnicos, dependências entre requisitos e preferências dos *stakeholders* seja implementado para a próxima versão do software (LETIER; STEFAN; BARR, 2014; GUEORGUIEV; HARMAN; ANTONIOL, 2009; LAMSWEERDE, 2009).

O risco considerado nesta Tese está associado à variabilidade das estimativas de valor de requisitos designadas por múltiplos *stakeholders*. Uma alta variabilidade nas estimativas indica alta probabilidade de insatisfação de um ou mais *stakeholders* se uma solução candidata for selecionada para a próxima versão do software. O impacto de tal insatisfação depende da perda de valor notada pelos *stakeholders* afetados e como eles percebem que a nova versão do projeto de software poderia ter atendido suas necessidades. Por isso, a inclusão de um ou mais requisitos com estimativas de valor muito distintas na próxima versão pode expor a empresa de software ao risco de insatisfação de alguns *stakeholders* e ao impacto negativo de perder o seu apoio ao projeto.

A ausência de apoio e envolvimento dos *stakeholders* no processo de análise, gerenciamento e seleção de requisitos é um dos fatores que afeta negativamente um projeto de software, sendo este um fator abordado por diversos trabalhos de autores especialistas na área de riscos relacionados a projetos de software (FAIRLEY; BOURQUE, 2014; GROUP, 2014; BECHIKH; SAID; GHÉDIRA, 2013; PRESSMAN, 2001; SOMMERVILLE, 2004; LAMSWEERDE, 2009; ASNAR; GIORGINI; MYLOPOULOS, 2011; FEATHER; CORNFORD, 2003; MOORES; CHAMPION, 1996; MCNEIL; FREY; EMBRECHTS, 2005; HAN; HUANG, 2007; HOODAT; RASHIDI, 2009; MELI, 1998; AMBER; SHAWOO; BEGUM, 2012; BANNERMAN, 2008; JORGENSEN, 2010).

Dessa forma, a inclusão de requisitos com estimativas de valor uniformes entre os *stakeholders* pode garantir que o valor acrescentado por estes requisitos seja entregue de modo mais uniforme a todos os interessados, com um risco reduzido de insatisfação. A próxima seção mostra com mais detalhes a formulação proposta quando há múltiplos *stakeholders* no processo de seleção de requisitos de software e como o risco da insatisfação dos *stakeholders* é abordado nesta Tese.

5.1 ABORDAGEM PARA O RA-MONRP COM MÚLTIPLOS STAKEHOLDERS

A Figura 5.1 apresenta o panorama para a abordagem RA-MONRP. Como pode ser visto nessa figura, no ambiente de desenvolvimento de software há a presença de múltiplos *stakeholders* e um conjunto de requisitos de software. Cada *stakeholder* possui um nível de importância para a empresa e cada requisito possui um custo de implementação e um valor atribuído por cada *stakeholder*. Por fim, existem dependências entre os requisitos que implicam em que um requisito seja implementado para que outro também o seja.

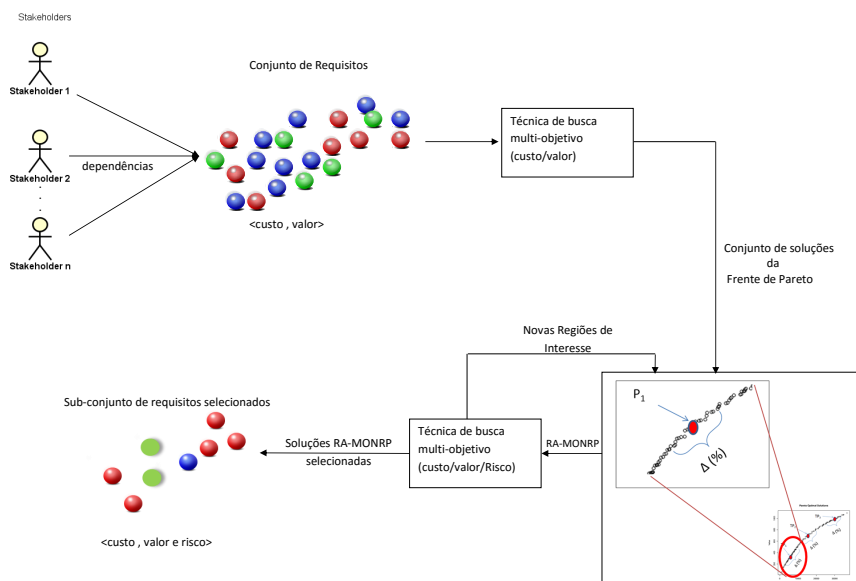


Figura 5.1: Visão Geral do RA-MONRP

A abordagem para seleção de requisitos de software inicia com a busca por soluções que levem em conta apenas os objetivos de custo e valor (MONRP). A formulação MONRP adotada nesse trabalho pode ser vista na Formulação 5.1. Nesse caso, deseja-se minimizar o custo (*cost*) e maximizar a média dos valores dos requisitos (*avgval*), levando-se em consideração as interdependências entre os requisitos ($x_j \Rightarrow x_i$). Logo, as entradas para as técnicas de busca multi-objetivo são os custos e média ponderada de valores dos requisitos, assim como as dependências entre os requisitos.

$$\left\{ \begin{array}{l} \text{Min } \sum_{i=1}^n \text{cost}_i \cdot x_i \\ \text{Max } \sum_{i=1}^n \text{avgval}_i \cdot x_i \end{array} \right. \quad \text{subject to } \bigwedge_{\langle R_i, R_j \rangle \in D} (x_j \Rightarrow x_i) \quad (5.1)$$

Como resultado do processo de busca para o MONRP é obtido um conjunto de soluções representado na forma de uma Frente de Pareto contendo os custos e valores médios para cada solução. Após obter essa Frente de Pareto, os *stakeholders* podem escolher algumas soluções (isto é, pontos P_i) baseados na combinação custo-valor médio aceitáveis de acordo com seus desejos e objetivos; por exemplo, considere que os *stakeholders* selecionaram os pontos P_1 , P_2 e P_3 na Figura 5.2.

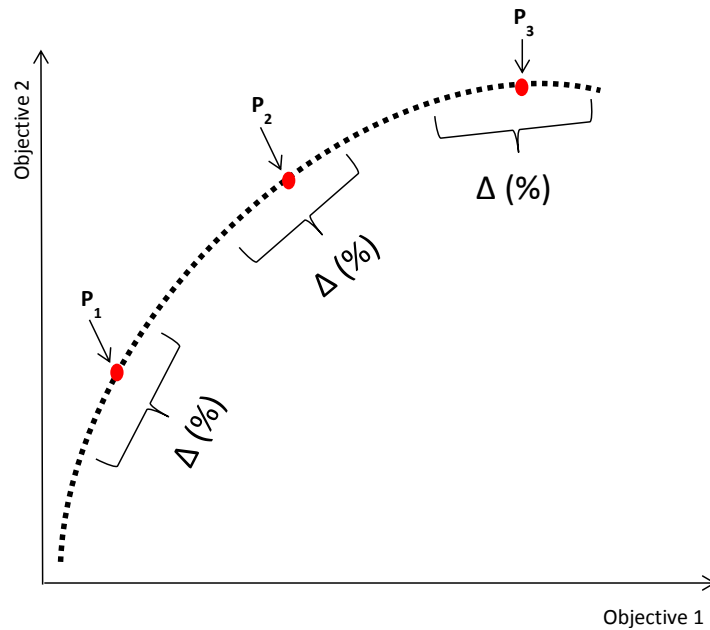


Figura 5.2: Regiões de Interesse para soluções P1, P2, P3 (PITANGUEIRA et al., 2016)

Uma vez que esses pontos são escolhidos, os *stakeholders* decidem sobre uma margem de tolerância para o custo e valor das soluções que levaram em consideração o risco de insatisfação, isto é, uma vizinhança para cada ponto selecionado é admitida e expressa em termos de uma variação percentual aceitável (Δ) para o custo e valor médio dos requisitos; por exemplo, suponha uma variação de 5% para custo e valor médio.

Escolhidas as regiões de interesse, o risco de insatisfação dos *stakeholders* pode ser integrado ao processo de otimização através de uma nova formulação multi-objetivo (RA-MONRP), descrita a seguir (PITANGUEIRA et al., 2016):

$$\left\{ \begin{array}{l} \text{Min} \sum_{i=1}^n cost_i \cdot x_i \\ \text{Max} \sum_{i=1}^n avgval_i \cdot x_i \\ \text{Min} \sum_{i=1}^n risk_i \cdot x_i \end{array} \right. \quad \text{subject to} \quad \bigwedge_{\langle R_i, R_j \rangle \in D} (x_j \Rightarrow x_i) \quad (5.2)$$

Como pode ser visto na formulação 5.2 do RA-MONRP, três objetivos estão em evidência: 1) somatório do custo dos requisitos (que deve ser minimizado); 2) somatório da média de valores dos requisitos (que deve ser maximizado); e 3) somatório do risco de insatisfação dos *stakeholders* (que deve ser minimizado). A restrição de dependência entre os requisitos continua expressa na relação $(x_j \Rightarrow x_i)$.

O fator de risco intrínseco associado com a variabilidade das estimativas dos *stakeholders* e possibilidade de insatisfação dos mesmos pode ser medido como a variância das estimativas de valor (Peter Goos, 2015; MCNEIL; FREY; EMBRECHTS, 2005), relacionando o peso dos *stakeholders* (w_j), o valor de importância para cada requisito (R_i) atribuído por cada *stakeholder* (S_j) e a média de valores dos requisitos ($avgval$). A Equação 5.3 mostra a formulação para o cálculo do risco.

$$risk_i = \sum_j w_j \cdot (valor(R_i, S_j) - avgval_i)^2 / n. \quad (5.3)$$

Para a formulação do RA-MONRP, as entradas no processo de busca são os custos, os valores médios e o risco de cada requisito e as dependências entre os requisitos. Alternativamente, o valor atribuído por cada *stakeholder* a cada risco poderia ser introduzido no processo de busca, permitindo o cálculo dos valor médio e do risco de cada requisito. Os limites selecionados (advindos da margem de tolerância) são incluídos na formulação RA-MONRP como restrições do problema.

Finalmente, um SMT *solver* é executado sobre a formulação RA-MONRP, gerando como saída uma Frente de Pareto em que cada solução possui um conjunto de requisitos, seu custo total, valor médio total e o risco total dessa solução (PITANGUEIRA et al., 2016). Com essa Frente de Pareto, os *stakeholders* podem começar o processo de seleção de uma solução para planejar a próxima versão do software.

Para analisar o impacto da redução do risco no custo e valor médio, cada solução produzida pelo processo de busca aplicado ao RA-MONRP é apresentada em um histograma que mostra a diferença de custo/valor médio de porcentagem para a solução previamente selecionada (P_i). O objetivo do histograma é fornecer aos *stakeholders* uma maneira de comparar as soluções RA-MONRP e dar suporte ao processo de tomada de decisão que levará à seleção final de requisitos para a próxima versão, considerando-se a redução de risco e seu impacto na otimização do custo e valor médio.

Para ilustrar essa abordagem, leve-se em consideração o caso de um software a ser desenvolvido, onde existem 50 requisitos, 4 stakeholders com graus de importância para a empresa variando de 1 – menos importante a 9 – muito importante. Cada requisito contém um custo associado, um valor de importância atribuído por cada stakeholder (1 – pouco importante a 9 – muito importante) e a relação de dependência, caso exista, com outros requisitos.

Após execução para o problema com custo e valor, os stakeholders se reuniriam e poderiam escolher uma ou mais soluções, e então estabeleceriam as margens de tolerância aceitáveis levando em consideração o orçamento disponível e a importância da solução. Por exemplo, se o orçamento naquele momento estiver limitado, os stakeholders podem selecionar uma solução no início da frente de Pareto, onde há opções com menor custo.

Escolhida a solução e a margem de tolerância (exemplo: $\Delta = 5\%$), o SMT solver é executado e as soluções na região de interesse são apresentadas comparando-se o percentual de custo, valor e risco de cada solução com a solução inicial escolhida. Um caso poderia ser a existência de uma solução na ROI que possua uma redução de custo em 1.96%, uma redução do risco em 11.65%, e uma perda de valor em 4.81% em relação à solução inicial selecionada como ponto de partida para a análise. Logo, essa nova solução seria mais interessante que a inicialmente escolhida pois apresenta menor custo, um risco menor representando uma maior concordância com a maioria dos stakeholders e uma perda de valor não tão significativa.

Esse processo decisório pode ser repetido ampliando-se as margens de tolerância ou restringindo-as, ou até que os stakeholders estejam satisfeitos e entrem em concordância de qual conjunto de requisitos será implementado para a próxima versão.

A próxima seção detalha as implementações das técnicas de busca que foram usadas nesse trabalho de Tese.

5.2 IMPLEMENTAÇÃO DAS TÉCNICAS DE BUSCA

A formulação proposta para o RA-MONRP foi implementada usando duas estratégias distintas visando obter um conjunto de soluções para o problema. Para avaliar o tempo de execução e qualidade das soluções obtidas, duas técnicas de busca determinísticas baseadas na *Satisfiability Modulo Theory*, Yices (DUTERTRE; MOURA, 2006) e Z3 (MOURA; BJØRNER, 2008), foram selecionadas e implementadas de acordo com os objetivos e restrições declarados na formulação do RA-MONRP.

Tanto o Yices como o Z3 são *solvers* determinísticos que utilizam a SMT para verificar se uma fórmula satisfaz ou não satisfaz uma ou mais condições apresentadas. Essas condições podem estar expressas em aritmética linear, quantificadores com ponto flutuante, lógica booleana, entre outros (DUTERTRE; MOURA, 2006). Por exemplo, dado um modelo matemático contendo funções e condições e expresso na notação do Yices ou Z3, os seguintes resultados podem ser obtidos: 1) ‘SAT’, quando o modelo satisfaz as condições; 2) ‘UNSAT’, quando o modelo não satisfaz as condições; e 3) ‘DESCONHECIDO’, quando o *solver* não foi capaz de decidir se as condições podem ser satisfeitas ou não (MOURA; BJØRNER, 2008).

Para comparar a escalabilidade das técnicas, o algoritmo NSGA-II (DEB et al., 2006),

uma meta-heurística bastante aplicada na área de SBSE¹, foi implementado para encontrar soluções para a abordagem proposta. Esse algoritmo difere dos SMT *solvers* (Yices e Z3) porque é uma técnica probabilística. Mais detalhes das implementações das técnicas de busca serão discutidos nas próximas subseções.

5.2.1 Implementação da Abordagem através de SMT

O pseudo-código do Algoritmo 1 mostra a implementação da abordagem para o Yices (PITANGUEIRA et al., 2016). Essa implementação visa encontrar soluções no espaço de busca para os objetivos de custo e valor. Os parâmetros de entrada no Algoritmo 1 são os custos e valores médios de cada requisito de software, assim como as dependências entre os requisitos (linha 1). A saída do Algoritmo 1 é o conjunto de soluções que satisfazem as condições do problema MONRP (linha 10).

Algorithm 1 Implementação do Yices para dois objetivos

```

1: Input:  $cost_i$  e  $avgval_i$  para cada requisito  $R_i$ ; dependências e preferências
2: Output: Frente de Pareto exata
3:  $lastcost = \min_i cost_i$ 
4:  $lastavgval = \min_i avgval_i$ 
5: while  $lastcost \leq totalcost$  do
6:   Adicione a restrição:  $\sum_{i=1}^n cost_i \cdot x_i = lastcost$  às entradas
7:   Adicione a restrição:  $\sum_{i=1}^n avgval_i \cdot x_i = lastavgval$  às entradas
8:    $yicesresult = \text{InvokeYices}()$ ;
9:   if  $yicesresult == \text{SAT}$  then
10:     Atualize a Frente de Pareto;
11:      $lastavgval = lastavgval + 1$ ;
12:   else
13:      $lastcost = lastcost + 1$ ;
14:   end if
15: end while

```

A implementação Yices percorre o espaço de solução fixando o custo mínimo ($lastcost$) e aumentando o valor ($lastavgvalue$) para encontrar o valor máximo para um determinado custo mínimo (linhas 3 e 4). Se uma solução satisfaz as condições da formulação (denominado como SAT), de acordo com o Yices (invocado na linha 5), os valores dos objetivos e subconjunto de requisitos dessa solução são atualizados e armazenados. Do contrário, isso significa que não existe qualquer solução que satisfaça as condições com o custo mínimo que está sendo analisado (denominado como UNSAT).

Quando uma solução é factível e satisfaz as condições impostas pelo problema, SAT é devolvido como resposta e a implementação aumenta $avgval$, de modo a tentar encontrar uma solução com o mesmo custo e maior valor (linhas 9 a 11). Quando UNSAT é devolvido como resposta, o custo é aumentado (linha 13) e a pesquisa é realizada novamente para encontrar soluções com maior custo e maior valor. O laço é interrompido

¹Vide resultados da revisão sistemática, no capítulo 2.

quando o custo total (por exemplo, o orçamento da empresa disponível) é atingido. Cada solução encontrada pelo Yices consiste no custo de implementação, o valor médio para os *stakeholders* e os requisitos contidos na solução.

Uma segunda implementação foi desenvolvida para a abordagem com base no Z3 e pode ser visualizada no pseudo-código do Algoritmo 2 (PITANGUEIRA et al., 2016). Essa implementação é similar ao Yices: os parâmetros de entrada são os custos e valores médios dos requisitos de software e suas dependências (linhas 2 e 5) e a saída é o conjunto de soluções (linha 14).

Algorithm 2 Implementação do Z3 para dois objetivos

```

1: (declare-fun  $R_1, \dots, R_n$  ( ) Bool)
2: (declare-fun  $costR_1, \dots, costR_n$ )
3: (declare-fun totalcost ( ) Int)
4: (declare-fun totalavgval ( ) Int)
5: (declare-fun  $avgvalR_1, \dots, avgvalR_n$ )
6: (declare-fun  $avgval_1, \dots, avgval_n$  ( ) Int)
7: (assert ( $\Rightarrow R_i R_j$ )  $\forall i, j \in D$ )
8: (assert (and (=  $costR_i$  (ite  $R_i$   $cost_i$  0)) ( $\Leftarrow$  0  $costR_i$ ) ( $\Leftarrow$   $costR_i$   $cost_i$ )))
9: (assert (and (=  $avgvalR_i$  (ite  $R_i$   $avgval_i$  0)) ( $\Leftarrow$  0  $avgvalR_i$ ) ( $\Leftarrow$   $avgvalR_i$   $avgval_i$ )))
10: (assert (cost =  $\sum_{i=1}^{i=n} cost_i$ ))
11: (minimize cost)
12: (assert (avgval =  $\sum_{i=1}^{i=n} avgval_i$ ))
13: (maximize avgval)
14: (set-option :opt.priority pareto)
15: (set-option :opt.print-model true)
16: (check-sat)

```

Como pode ser visto no Algoritmo 2, as funções objetivo (minimização do custo e maximização do valor médio) podem ser diretamente explicitadas em um modelo Z3 (linhas 10 a 13). Neste modelo, cada requisito é uma variável booleana R_1, \dots, R_n contendo uma variável associada de custo ($costR_i$) e um valor de custo ($cost_i$), o mesmo acontecendo para os valores de cada requisito de software. As relações entre os requisitos (interdependências), são expressas como uma implicações lógicas. A seguinte expressão é usada para associar um custo diferente de zero aos requisitos selecionados: (assert (and (= $costR_i$ (ite R_i $cost_i$ 0)) (\Leftarrow 0 $costR_i$) (\Leftarrow $costR_i$ $cost_i$))).

Se o requisito R_i é selecionado (*ite = if-then-else*), o custo desse requisito é igual a $cost_i$; caso contrário, é zero. Expressões semelhantes são usadas para associar um valor diferente de zero para os requisitos selecionados (linha 9). O modelo Z3 pode ser facilmente estendido com variações, como a inclusão de mais objetivos, especificando uma região de interesse no espaço de busca e adicionando novas restrições.

Apesar de ambos os *solvers* utilizarem SMT como base, o Z3 oferece uma vantagem em relação ao Yices por conter um módulo interno pré-implementado para a obtenção da

Frente de Pareto (linha 14, Algoritmo 2), algo até então não oferecido pelo Yices.

5.2.2 Implementação através da Meta-Heurística NSGA-II

O NSGA-II (DEB et al., 2002) é um algoritmo que suporta problemas multi-objetivo e se baseia no elitismo para encontrar e manter as melhores soluções na Frente de Pareto. Sua complexidade está na ordem $O(MN^2)$, onde M é o número de objetivos que o problema abordado apresenta e N é o tamanho da população. O NSGA-II atenua dois problemas que algoritmos evolucionários multi-objetivo anteriores apresentavam: complexidade na ordem $O(MN^3)$ e a falta de elitismo. O NSGA-II foi adaptado ao problema de seleção de requisitos usado na abordagem proposta por esse trabalho de Tese.

Em linhas gerais, o NSGA-II usa um algoritmo para buscar por soluções próximas à Frente de Pareto (Algoritmo 3) e outro algoritmo para buscar por soluções bem distribuídas no espaço de busca (Algoritmo 4). A população (conjunto de soluções que estão sendo analisadas) é classificada usando a relação de não-dominância em várias frentes de Pareto. A primeira frente possui soluções que não são dominadas por nenhuma outra solução. A segunda frente possui soluções que são dominadas apenas por aquelas da primeira frente e assim sucessivamente até a última frente. O pseudo-código do Algoritmo 3 mostra a implementação do NSGA-II para a busca por soluções ótimas próximas à Frente de Pareto (DEB et al., 2002). A entrada do algoritmo é um conjunto de requisitos, com seus custos, valores médios e as dependências entre os requisitos. Como saída são produzidas as possíveis soluções para cada Frente de Pareto.

O algoritmo 3 atribui a cada solução um valor de ajuste de acordo com seu nível de não-dominância. Dessa forma, as soluções em melhores frentes de Pareto recebem valores de ajuste mais elevados. Todas as soluções na primeira Frente de Pareto terão seus contadores de dominância iguais a zero (linhas 2 e 3). Para cada solução p com $n_p=0$, cada membro q do conjunto S_p é visitado e se reduz sua dominância a 1 (linhas 4 a 13). Por conseguinte, caso o contador de dominância para qualquer membro q se tornar zero, esse membro será posto em uma lista separada Q (linha 17). Cada membro nessa lista pertence à segunda frente de soluções. Esse procedimento continua com cada membro Q e a terceira frente é identificada. O processo se repete até que todas as Frentes de Pareto sejam identificadas.

A população da próxima rodada do algoritmo será composta pelos indivíduos componentes das primeiras Frentes de Pareto, até o limite estabelecido para o tamanho da população. O algoritmo NSGA-II utiliza uma medida de distância para fornecer uma estimativa da densidade de soluções que pertencem a uma Frente de Pareto. Esta medida é um parâmetro utilizado para promover a diversidade dentro da população: caso uma Frente de Pareto seja maior do que o espaço disponível para formar a população da próxima geração, os indivíduos que serão levados desta frente para a população serão selecionados segundo este critério. Ele atribui um valor maior de aptidão às soluções que estão a uma maior distância de outras soluções na Frente de Pareto do que a soluções próximas de outras soluções da frente, levando a uma seleção de várias regiões da frente para a nova população ao invés de uma concentração em uma região. Seu código pode ser analisado no Algoritmo 4 em que as entradas são as soluções (ou indivíduos) de cada

Algorithm 3 NSGA-II (fast-non-dominated-sort) (DEB et al., 2002)

```

1: for each  $p \in P$  do
2:    $S_p = 0$  // conjunto de soluções dominadas pela solução  $p$ 
3:    $n_p = 0$  // contador de dominância
4:   for  $q \in P$  do
5:     if ( $p \prec q$ ) then // se o indivíduo  $p$  domina o indivíduo  $q$ 
6:        $S_p = S_p \cup q$  // adiciona  $q$  ao conjunto de soluções dominadas por  $p$ 
7:     else if ( $q \prec p$ ) then
8:        $n_p = n_p + 1$  // incrementa o contador de dominância de  $p$ 
9:     end if
10:  end for
11:  if  $n_p == 0$  then //  $p$  pertence à primeira Frente
12:     $p_{rank} = 1$ 
13:    Let  $F_1 = F_1 \cup p$ 
14:  end if
15:  Set  $i=1$  // inicializa o contador de Frentes
16:  while  $F_i \neq 0$  do
17:     $Q = 0$  // armazena os membros da próxima Frente
18:    for each  $p \in F_i$  do
19:      for each  $q \in S_p$  do
20:         $n_q = n_q - 1$ 
21:        if  $n_q == 0$  then // indivíduo  $q$  pertence a próxima frente
22:           $q_{rank} = i + 1$ 
23:           $Q = Q \cup q$ 
24:        end if
25:      end for
26:    end for
27:     $i = i + 1$ 
28:     $F_i = Q$ 
29:  end while
30: end for

```

Frente de Pareto e a saída é o resultado comparativo (ou diversidade) entre as soluções de cada Frente de Pareto.

Algorithm 4 NSGA-II (Crowding-distance-assignment) (DEB et al., 2002)

```

1:  $l = |C|$  // número de soluções em  $\Gamma$ 
2: for each individual (i) do
3:   Set  $C_i$  // inicializa a distância
4:   for each objective  $m$  do
5:     Let  $C_i = \text{sort}(C, m)$  // ordena usando cada valor de objetivo
6:      $C_1 = C_l = \infty$  // nesse caso, os pontos da fronteira são sempre selecionados
7:   end for
8:   for each  $i = 2$  to  $(C - 1)$  do // para todos os outros pontos
9:      $C_i = C_i + (C_{[i+1]} \cdot m - C_{[i-1]} \cdot m) / (f_m^{\max} - f_m^{\min})$ 
10:  end for
    //  $C_i \cdot m$  corresponde a  $m_{th}$  função objetivo para o individuo  $i$ 
    //  $f_m^{\max}$  e  $f_m^{\min}$  são os valores máximo e mínimo da  $m_{th}$  função objetivo
11: end for

```

Com as implementações dos Algoritmos 3 e 4, o laço principal do NSGA-II pode ser executado (Algoritmo 5). A entrada é composta pelos custos, valores e dependências dos requisitos que farão parte de uma população inicial (linha 2) e a saída é a Frente de Pareto (linha 6 a 15). Inicialmente, uma população P_0 de tamanho N é gerada aleatoriamente e ordenada com base na relação de não-dominância, formando várias Frentes de Pareto. Para cada solução é atribuído um nível igual ao seu nível de não-dominância, isto é, 1 corresponde ao melhor nível, 2 é o próximo melhor nível, e assim por diante. Em seguida, os operadores de seleção por torneio binário, recombinação (*crossover*) e mutação são aplicados para criar uma nova população (Q_0) de tamanho N . O Algoritmo 5 mostra a implementação para o laço principal do NSGA-II.

É importante salientar que no laço principal do NSGA-II um operador de torneio binário foi utilizado, mas o critério de seleção para participação da nova população é baseado no operador de dominância \prec_n . Em relação à complexidade do algoritmo em uma iteração, as operações básicas no pior caso são as seguintes (DEB et al., 2002): a) ordenação para a não-dominância é $O(M(2N)^2)$, b) para o cálculo da distância é $O(M(2N)\log(2N))$, e c) para a ordenação em \prec_n é $O(2N\log(2N))$; perfazendo uma complexidade geral do algoritmo na ordem $O((MN)^2)$.

5.3 CONSIDERAÇÕES FINAIS

A abordagem proposta nesse capítulo mostra o processo de otimização para o RAMONRP, desde a formulação do MONRP (custo e valor) até a formulação do RA-MONRP (custo, valor e risco). Também foram apresentadas e discutidas as técnicas SMT e a meta-heurística NSGA-II que dão apoio ao processo de busca para obtenção de soluções. Essa abordagem oferece suporte à decisão na seleção de requisitos de software que minimizem o risco de insatisfação dos *stakeholders*, sendo que o processo de busca pode ser feito em

Algorithm 5 NSGA-II - Main Loop (DEB et al., 2002)

```

1: while  $t \leq N$  do
2:    $R_t = P_t \cup Q_t$  // combina as populações pai e descendente
3:    $C = \text{fast-non-dominated-sort}(R_t)$ 
4:   //  $C = (C_1, C_2, \dots, C_i)$  são todas as frentes de Pareto não dominadas de  $R_t$ 
5:    $P_{t+1} = \emptyset$ 
6:    $i = 1$ 
7:   while  $|P_{t+1}| + |C_i| \leq N$  do
8:     crowding-distance-assignment ( $C_i$ )
9:      $P_{t+1} = P_{t+1} \cup C_i$  // inclui a  $i$ th frente não dominada na população pai
10:     $i = i + 1$  // checa a próxima frente para a inclusão
11:  end while
12:  Sort ( $C_i, \prec_n$ ) // ordena de forma decendente
13:   $P_{t+1} = P_{t+1} \cup C_i[1 : (N - |P_{t+1}|)]$ 
14:   $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ 
15:  // usa seleção, cruzamento e mutação para criar a nova população  $Q_{t+1}$ 
16:   $t = t + 1$  // incrementa o contador de geração
17: end while

```

regiões de interesse distintas, de acordo com os desejos dos *stakeholders*.

De posse da abordagem proposta para o RA-MONRP e das técnicas de busca selecionadas para a otimização, o próximo capítulo apresenta a metodologia utilizada na experimentação para validação do RA-MONRP, os testes realizados com a abordagem e as técnicas de busca e os procedimentos adotados na Tese para a avaliação e validação dos resultados.

AVALIAÇÃO

6.1 METODOLOGIA PARA A EXPERIMENTAÇÃO DA PROPOSTA

Este capítulo apresenta os estudos experimentais projetados e executados para avaliar diferentes aspectos da formulação RA-MONRP. Todo o procedimento para a realização dos experimentos nesse trabalho de Tese segue a metodologia da engenharia de software experimental proposta por Wohlin et al. (WOHLIN et al., 2012) e Wieringa (WIERINGA, 2014). O processo de experimentação adotado na Tese obedece ao seguinte roteiro (WOHLIN et al., 2012; WIERINGA, 2014):

1. Planejamento dos experimentos;
2. Operação e execução dos experimentos;
3. Análise e interpretação dos resultados;
4. Publicação dos resultados e conclusões.

A Tabela 6.1 mostra as fases sequenciais da Tese, com o método e atividades correspondentes.

Tabela 6.1: Estágios da Pesquisa desenvolvidos na Tese

FASE	ATIVIDADE	MÉTODO
1	Gerar objetivos da pesquisa	Revisão e Mapeamento Sistemático
2	Formalização do Problema	Revisão e Mapeamento Sistemático
3	Elaborar diretrizes da pesquisa	Planejamento dos experimentos
4	Executar experimentos	Operação dos experimentos
5	Interpretar e descrever os resultados obtidos com os experimentos	Análise e escrita dos resultados

A primeira fase visou identificar os objetivos que esse trabalho deveria alcançar. Para se ter uma visão completa do assunto em questão, Wohlin et al (WOHLIN et al., 2012) afirmam que a condução de uma revisão sistemática é o melhor instrumento para se visualizar e compreender os fatos mais abordados no assunto estudado, observar as lacunas de pesquisa e identificar as possibilidades de trabalhos futuros. A revisão sistemática sobre seleção de requisitos de software ofereceu o embasamento teórico e técnico para se identificar o cenário de pesquisa e, por conseguinte, desenvolver o objetivo geral e os objetivos específicos anteriormente descritos.

De posse dos objetivos da pesquisa, a segunda fase deste trabalho focou na formalização do problema. A formalização é uma das tarefas essenciais porque estabelece o modelo de otimização que servirá como diretriz para o processo de experimentação. Nessa fase, a formulação multi-objetivo para o risco (RA-MONRP) foi desenvolvida e, mais uma vez, a revisão sistemática foi importante para direcionar a nova formulação, diferenciando-a de trabalhos relacionados.

Com os objetivos definidos (fase 1) e a formalização do RA-MONRP (fase 2) foi possível estabelecer as diretrizes da pesquisa. Nessa fase, de acordo com Wieringa (WIERINGA, 2014), devem ser identificadas as questões de pesquisa (RQ) a serem respondidas. Três questões de pesquisa foram identificadas: a primeira associada à possibilidade de se encontrar soluções que possam reduzir o risco da insatisfação do usuário com pequeno impacto nos demais objetivos; a segunda relativa à escalabilidade das técnicas aplicadas, com relação ao seu tempo de execução; finalmente, a terceira relativa à qualidade das soluções obtidas através das diferentes técnicas utilizadas na otimização.

Com esses elementos é possível elaborar o **planejamento dos experimentos** (fase 3) para validar as técnicas propostas (WOHLIN et al., 2012). Nessa etapa foram selecionados os dados a serem utilizados nos experimentos, as possibilidades de execução dos mesmos, o levantamento das questões de pesquisa e as ameaças à validade dos resultados observados. Para fins de proximidade com o mundo real, foram utilizados apenas dados reais de projetos de software nos experimentos.

Apesar da dificuldade de se obter dados reais, a abordagem proposta foi avaliada em dois conjuntos de dados de projetos de software. O primeiro (*dataset-1*) é baseado em um projeto para criação de uma ferramenta de processamento de texto (KARIM; RUHE, 2014) e possui 50 requisitos e 4 *stakeholders*. O segundo (*dataset-2*) tem 25 requisitos e 9 *stakeholders* e está relacionado ao *ReleasePlanner*TM, um software de apoio à decisão para planejamento estratégico de versões de software ao longo do seu ciclo de vida¹.

Esses dois conjuntos de dados incluem todas as informações necessárias para o estudo aqui proposto: estimativas de custo, precedências diretas e dependências entre os requisitos e informações dos *stakeholders* sobre o valor esperado de cada requisito. Além disso, os dois conjuntos de dados escolhidos incluem vários *stakeholders* que atribuíram diferentes níveis de importância para cada requisito, destacando suas diferentes percepções e desejos, às vezes em desacordo entre eles.

De posse dos *dataset-1* e *dataset-2*, houve um tratamento de dados para identificar os custos de cada requisito, verificar os valores médios de cada requisito de acordo com as

¹<https://www.releaseplanner.com>

atribuições dos *stakeholders*, rastrear as dependências entre os requisitos e desenvolver os cálculos do risco de acordo com a equação 5.3, apresentada no Capítulo 5.

Após o tratamento dos dados, uma implementação de algoritmo exaustivo foi desenvolvida para verificar o tempo de execução e quantidade de requisitos que uma técnica de força-bruta suportaria. No entanto, o algoritmo exaustivo (força-bruta) não foi capaz de resolver o problema através da abordagem proposta, pois o problema apresenta explosão combinatorial e o algoritmo exaustivo não consegue tratar o número de requisitos relacionados aos *dataset*. Devido a isso, foram selecionadas técnicas de busca mais eficientes e usadas na comunidade científica de SBSE e da pesquisa operacional (PITANGUEIRA; MACIEL; BARROS, 2015; PITANGUEIRA et al., 2016). Yices e Z3 foram técnicas exatas selecionadas por serem amplamente utilizadas em problemas da área de pesquisa operacional (De Moura; RNER, 2009; MOURA; BJØRNER, 2008; DUTERTRE; MOURA, 2006; HILLIER; LIEBERMAN, 2004) e o NSGA-II por ser uma meta-heurística validada por pesquisadores de SBSE (ZHANG; HARMAN, 2010; HARMAN, 2007).

Após essa seleção, as técnicas de busca foram implementadas de acordo com as especificações dos dados. Para as técnicas SMT, os valores associados a cada requisito (custo, valor médio e risco) foram escritos diretamente no código fonte. Para a meta-heurística, cada *dataset* foi representado em uma planilha com os valores envolvidos e o algoritmo recebeu esta planilha como informação de entrada.

Tendo o tratamento dos dados e as implementações desenvolvidas, a próxima fase foi a execução das técnicas de busca (fase 4) para a obtenção dos resultados. Os experimentos foram executados na mesma configuração de máquina, um MacBook Pro, OSX Yosemite, core™ i7 CPU 2.50 GHz, 16 GB RAM. Todas as técnicas de busca produziam como saída uma planilha contendo os resultados. Por exemplo, no MONRP, cada linha da planilha continha o subconjunto de requisitos (uma solução da Frente de Pareto) com o custo total e valor médio total da solução.

Seguindo a abordagem proposta, com as planilhas de resultados bi-objetivo para os *datasets-1 e 2* foram obtidas as Frentes de Pareto utilizando o software estatístico R². Selecionados os pontos de referência para as regiões de interesse (P_i), uma nova implementação das técnicas foi desenvolvida para conter o risco como novo objetivo e margens de tolerância para custo e valor como restrições do problema. Mais uma vez, os resultados do novo ciclo de otimização foram apresentados em planilhas e representados em histogramas. Também nessa fase, em cada execução de experimento foram coletados o tempo de execução em segundos e a quantidade de soluções obtidas na Frente de Pareto.

Para medir o desempenho das técnicas SMT, uma execução é suficiente por se tratar de técnicas determinísticas. Em relação ao NSGA-II, para medir seu desempenho médio, essa meta-heurística foi executada 100 vezes, uma vez que o seu comportamento é não-determinístico (ARCURI; BRIAND, 2011). Os seguintes parâmetros foram usados para configurar o NSGA-II: probabilidade de cruzamento de $P_c = 80\%$ e probabilidade de mutação $P_m = 1/n$, onde n é o número de variáveis de decisão booleanas (ou seja, o tamanho do cromossomo na representação de vetor de bits de uma solução ou o número de requisitos disponíveis para seleção na próxima versão do software).

²<https://www.r-project.org/>

Executados os experimentos e obtidos os resultados, a fase seguinte (fase 5) teve como objetivo interpretar os resultados obtidos durante o experimento, usando tanto uma abordagem quantitativa quanto qualitativa. Nessa etapa, os histogramas e tabelas consolidadas (tempo, quantidade de soluções e percentagens) foram estudados e analisados, a fim de responder às questões de pesquisa levantadas.

6.2 PROCEDIMENTO PARA AVALIAÇÃO E VALIDAÇÃO DOS RESULTADOS

O projeto experimental adotado para a validação desse trabalho de Tese consiste na definição de questões de pesquisa (RQ) e a forma como cada avaliação (E) destas questões deve ser conduzida, o que inclui o tratamento sob avaliação, as medidas a serem tomadas e as análises a ser aplicadas. A Tabela 6.2 mostra os principais elementos do projeto experimental. Foram definidas três questões de pesquisa e as correspondentes estratégias para executar a avaliação experimental.

Tabela 6.2: Quadro Conceitual da Pesquisa

QUESTÕES DE PESQUISA (RQ_n)	AVALIAÇÃO (E_n)
RQ_1 : É possível encontrar soluções que reduzam o risco de insatisfação dos <i>stakeholders</i> com impacto mínimo no custo e valor dos requisitos?	<p>E_1: Dada uma Frente de Pareto (P) com dois objetivos (custo/valor), selecionar uma região de interesse (ROI) e calcular a nova Frente de Pareto para três objetivos (custo, valor e risco) dentro do ROI. Medir a variação percentual do custo, valor e risco das novas soluções com respeito às soluções originais em P.</p> <p><i>Tratamento 1</i>: Uso de margens de tolerância iguais e fixas para custo e valor;</p> <p><i>Tratamento 2</i>: Uso de uma margem de tolerância livre e outra fixa;</p> <p><i>Tratamento 3</i>: Uso de margens de tolerância diferentes para cada objetivo.</p>
RQ_2 : Qual a escalabilidade da abordagem quando o cenário cresce em número de requisitos, em número de dependências ou em ambos?	E_2 : Executar experimentos aumentando o número de requisitos e dependências. Comparar o tempo de execução requerido pelas técnicas SMT e meta-heurística para encontrar as soluções.
RQ_3 : Quão próximas são as soluções encontradas pela meta-heurística e pelo SMT?	E_3 : Medir a diferença percentual entre o conjunto de soluções de Pareto obtido pelo SMT e pela meta-heurística.

A seguir, para cada questão de pesquisa, é discutido o que foi investigado e o pro-

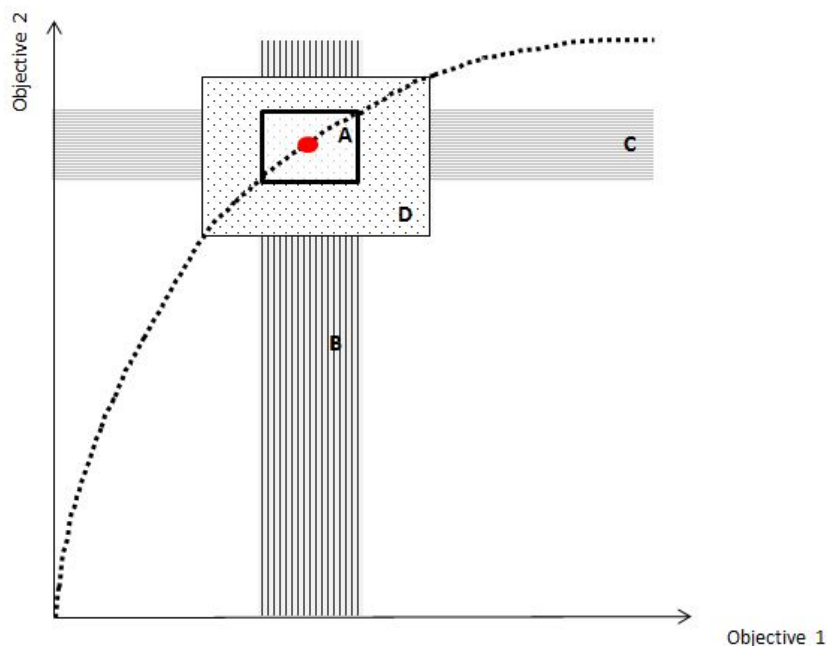


Figura 6.1: Ilustração do cenário da abordagem para diferentes ROIs

cesso de experimentação e avaliação adotado é apresentado em detalhes. Para facilitar o acompanhamento do leitor, essa explanação seguirá a ordem da Tabela 6.2.

6.2.1 RQ1. É possível encontrar soluções que reduzam o risco de insatisfação dos stakeholders com impacto mínimo no custo e valor dos requisitos?

Essa questão de pesquisa avalia a existência de soluções ótimas que possam reduzir o impacto negativo da insatisfação dos *stakeholders* com uma solução em decorrência da variação das estimativas de valor atribuído a cada requisito pelos diferentes *stakeholders*.

Com o intuito de responder a esta questão de pesquisa, a abordagem proposta no capítulo anterior (seleção de Regiões de Interesse) foi aplicada para cada tipo de tratamento sugerido na Tabela 6.2. A Figura 6.1 mostra uma visão geral e os principais elementos da abordagem proposta para a obtenção de soluções na Frente de Pareto.

Uma vez que os pontos que caracterizam as regiões de interesse sejam selecionados, os *stakeholders* decidem sobre uma margem de tolerância para o custo e para o valor a fim de avaliar o risco que cada solução apresenta nesse intervalo. Em outras palavras, uma vizinhança contendo um conjunto de soluções é gerada com base nessa margem de tolerância, que é expressa em termos de uma variação percentual (Δ) de custo e valor. As margens de tolerância podem ocorrer de acordo com os exemplos a seguir:

1. Tratamento 1
 - (a) Margens fixas e iguais para os dois objetivos (Região A)
2. Tratamento 2

- (a) Margem fixa para custo e livre para valor (Região B)
- (b) Margem livre para custo e fixa para valor (Região C)

3. Tratamento 3

- (a) Margens de tolerância fixas, mas diferentes para cada objetivo (Região D)

No caso do Tratamento 1, em que as margens são fixas e iguais para custo e valor, pode-se, por exemplo, especificar uma margem de tolerância (Δ) de 5% para o custo e a mesma margem de 5% para o valor.

No que diz respeito ao Tratamento 2, sempre um dos objetivos não sofrerá nenhuma limitação. Por exemplo, pode-se deixar o valor variar livremente, enquanto o custo pode ser fixado em uma margem de tolerância de 3% (item 2.a). O contrário também pode ocorrer, em que o valor pode ter $\Delta = 3\%$ e o custo não sofrer limitação (item 2.b).

Por fim, o Tratamento 3 é uma variante do Tratamento 1. Nesse caso, as margens de tolerância são fixas, porém diferentes entre os objetivos. Por exemplo, podemos estabelecer que $\Delta = 5\%$ para o custo, enquanto $\Delta = 4\%$ para o valor.

Qualquer uma dessas situações pode ser incluída na formulação com três objetivos (custo, valor médio e risco) do RA-MONRP como restrições do problema. Finalmente, o SMT *solver* é executado para a formulação do RA-MONRP com as restrições impostas de acordo com o tratamento e as soluções (representadas como triplas $\langle cost, avgval, risk \rangle$) ótimas são identificadas para cada região de interesse indicada pelos *stakeholders*, que então podem iniciar um processo de tomada de decisão para planejar a próxima versão do software, escolhendo uma dentre as soluções ótimas retornadas pelo algoritmo.

A diferença de se usar um SMT *solver* ao invés de um algoritmo meta-heurístico é que o primeiro retorna a solução exata para o problema (isto é, a Frente de Pareto ótima). Quando as instâncias do problema aumentam, seja por acréscimo do número de requisitos ou de dependências entre eles, tornando o SMT *Solver* inaplicável em tempo aceitável, algoritmos meta-heurísticos podem ser utilizados a fim de obter uma Frente de Pareto aproximada, ou seja, sem a garantia de que todas as soluções sejam ótimas de fato.

6.2.2 RQ2. Qual a escalabilidade da abordagem quando o cenário cresce em número de requisitos, em número de dependências ou em ambos?

A segunda questão de pesquisa (RQ2) está relacionada com a escalabilidade das técnicas de busca utilizadas para encontrar soluções que servirão de apoio para a escolha das Regiões de Interesse para o RA-MONRP.

Para testar a escalabilidade da abordagem proposta, foram investigados o tempo de execução necessário para obter as frentes exatas de Pareto usando SMT *solver* e a frente de Pareto aproximada através do algoritmo NSGA-II. Para esse caso, não foi considerado o SMT Yices, uma vez que ficou demonstrado, em um trabalho anterior, que esse SMT apresenta um aumento acentuado no tempo de execução quando o número de requisitos aumenta (PITANGUEIRA et al., 2016).

Com o objetivo de manter as características intrínsecas dos dados usados para experimentação, foi desenvolvido um procedimento para gerar novas instâncias para os estudos

de escalabilidade, tendo como princípio um fator de crescimento baseado na relação requisitos/dependências dos dados originais e, posteriormente, de cada nova instância gerada (PITANGUEIRA et al., 2016). Para cada *dataset*, foi calculada a proporção entre o número de requisitos e o número de dependências (*average_factor*) e um fator de incremento do número de dependências (*inc_deps*) de acordo com o número de requisitos desejado na instância que será gerada.

Para ilustrar o procedimento para obtenção desses fatores, considere um grafo com quatro requisitos ($| Reqs | = 4$) e duas relações de dependência entre estes requisitos ($| Deps | = 2$). Então, a proporção requisitos/dependências (*average_factor*) deste grafo pode ser calculada através da fórmula abaixo:

$$average_factor = | Deps | / | Reqs | = 2/4 = 0,5 \quad (6.1)$$

Depois, caso se decida incluir, por exemplo, mais dois requisitos à nova instância, o número correspondente de dependências a ser acrescentado pode ser obtido multiplicando-se a nova quantidade de requisitos (*NewReqs*) pelo (*average_factor*), da seguinte forma:

$$inc_deps = | NewReqs | * | average_factor | = 2 * 0,5 = 1 \quad (6.2)$$

Assim, a nova instância terá 6 requisitos (4 requisitos vieram do grafo original e dois adicionados) e três dependências (2 vieram do original e um do fator de incremento de dependências *inc_deps*). Um detalhe a ser ressaltado é que o fator *inc_deps* deve ser um número inteiro, uma vez que não há a possibilidade de incluir um número de dependências fracionado. Então, pode ser necessária uma aproximação, como no caso em que o fator de incremento de dependências tenha seu valor calculado como 0,88 (*inc_deps* = 0,88). Neste caso, recomenda-se aproximar o valor para o número inteiro mais próximo (no caso, a inclusão de uma nova dependência).

A geração de novas instâncias deve respeitar este processo de cálculo para a inclusão de novos requisitos e dependências. Em outras palavras, se instâncias de tamanho crescente forem geradas em sequência, como é o caso nesta análise de escalabilidade, esse processo se repete tomando como ponto de partida a instância imediatamente anterior.

Após a obtenção da quantidade de requisitos e dependências a serem incluídos na nova instância, os grafos correspondentes são gerados com o auxílio da ferramenta geradora de grafos *yed Graph Generator*TM ³.

O cenário acima foi ilustrado para o procedimento de escalabilidade observando a expansão concomitante do número de requisitos e dependências. Há ainda mais duas situações: a) quando se deseja aumentar apenas o número de requisitos; e b) quando apenas o número de dependências é aumentado. Em ambos os casos, a geração de novas instâncias, maiores em relação à original, pode ser feita diretamente aplicando uma taxa constante de crescimento, uma vez que apenas um dos atributos (requisitos ou dependências) sofrerá alteração a cada rodada. Tomando-se o grafo do exemplo anterior,

³<https://www.yworks.com/products/yed>

pode-se gerar cada nova instância acrescentando um determinado fator r de crescimento somente do número de requisitos. Por exemplo, para $r = 2$, a próxima instância terá seis requisitos, a subsequente 8, e assim por diante.

O mesmo critério é adotado quando se deseja o crescimento apenas do número de dependências. Escolhendo-se um fator $d = 1$ para o crescimento de cada nova instância, o exemplo anterior poderia ter uma nova instância com três dependências, a próxima com quatro dependências, e assim por diante.

6.2.3 RQ3. Quão próximas são as soluções encontradas pela implementação da meta-heurística e da implementação baseada em SMT?

Essa questão de pesquisa toma como base a configuração original dos *datasets* e os resultados obtidos com as instâncias artificiais geradas através do processo apresentado na seção anterior. Todas as instâncias são consideradas para avaliar a qualidade das soluções encontradas pela meta-heurística em diferentes cenários, sempre em comparação com as Frentes de Pareto ótimas encontradas pelo SMT para as mesmas instâncias.

A análise para essa questão de pesquisa não requer novas implementações ou geração de novas instâncias, uma vez que ambas foram geradas nas experimentações anteriores. Para quantificar a qualidade das soluções obtidas pelas técnicas de busca, o seguinte procedimento experimental foi adotado:

1. O SMT produz uma Frente de Pareto para cada instância;
2. NSGA-II produz 100 Frentes de Pareto para cada instância, em 100 execuções, uma vez que essa técnica é não-determinística;
3. Para cada solução em cada Frente de Pareto produzida pelo NSGA-II, a solução com custo mais próximo na Frente de Pareto encontrada pelo SMT é selecionada e a diferença percentual de custo correspondente é armazenada;
4. Para cada solução em cada Frente de Pareto produzida pelo NSGA-II, a solução com valor mais próximo na Frente de Pareto encontrada pelo SMT é selecionada e a diferença percentual de valor correspondente é armazenada.

Como explicado anteriormente, o NSGA-II pode produzir soluções iguais ou próximas da Frente de Pareto real (encontrada pelo SMT), incluindo assim tanto soluções pertencentes à Frente de Pareto quanto soluções sub-ótimas excedentes. Sendo assim, da mesma forma que é possível verificar a proximidade das soluções sub-ótimas do método probabilístico em relação ao método determinístico, é também possível comparar o número de soluções estritamente iguais que a meta-heurística obteve em relação ao SMT.

Para se realizar essa comparação, o conjunto de soluções obtido pelo Z3 foi confrontado com cada conjunto obtido pelo NSGA-II. Na comparação entre a Frente de Pareto do SMT e uma Frente de Pareto do NSGA-II, se verificou a quantidade de soluções iguais ao resultado do Z3 que foram obtidas pelo NSGA-II. Por fim, é feito o cálculo do percentual de similaridade entre as frentes, levando como referência a totalidade das soluções obtidas

pela técnica determinística. Por exemplo, se o Z3 obtivesse 100 soluções e o NSGA-II obtivesse 130, das quais 80 soluções são iguais ao Z3, o NSGA-II obteria 80% de aproximação em relação à Frente de Pareto do Z3. Nesse exemplo e todos os demais casos possíveis, o cálculo da percentagem de proximidade segue como mostrado abaixo:

$$perc_prox = \frac{|Sol_{NSGA-II} \cap Sol_{Z3}|}{|Sol_{Z3}|} = 80/100 = 80\% \quad (6.3)$$

onde $perc_prox$ é o percentual de soluções exatamente iguais entre as técnicas, sendo obtido através da divisão do número de soluções iguais do NSGA-II em relação ao Z3 dividido pelo total de soluções obtidas pelo Z3.

Assim, é possível obter um panorama comparativo da qualidade das soluções encontradas pelas duas técnicas utilizadas sob duas perspectivas: a primeira delas, em relação à igualdade de soluções obtidas pela técnica probabilística em relação à técnica determinística; a segunda, em relação à proximidade das soluções excedentes (sub-ótimas) obtidas pela técnica probabilística em relação às soluções ótimas do SMT. Para execução dessas comparações foram usados os softwares Excel 365 for Mac⁴ e DiffMerge⁵.

A próxima seção apresenta os resultados obtidos para cada questão de pesquisa proposta.

6.3 RESULTADOS E AVALIAÇÃO

6.3.1 Resultados para a Redução do Risco de Insatisfação dos Stakeholders - RQ1

6.3.1.1 Tratamento 1 Como descrito nas seções anteriores, a abordagem proposta foi inicialmente testada através do uso de dois SMT: Yices e Z3. As Tabelas 6.3 e 6.4 mostram os resultados obtidos para o primeiro passo descrito na abordagem proposta, isto é, o tempo total (em segundos) e o número de soluções nas Frentes de Pareto para a formulação bi-objetivo (MONRP). As Figuras 6.2 e 6.3 mostram as Frentes de Pareto obtidas com o Z3 para cada *dataset*.

Tabela 6.3: Resultados - *dataset-1*

Solver	Tempo(s)	Soluções
Yices	538.658,00	385
Z3	183.320,29	285

Tabela 6.4: Resultados - *dataset-2*

Solver	Tempo(s)	Soluções
Yices	2.939,00	146
Z3	81,86	143

Através dos resultados das tabelas acima, pode ser percebido que o Yices produz mais soluções que o Z3, em ambos os casos. Uma investigação realizada sobre estas soluções mostrou que o Yices inclui soluções *weak Pareto* em suas Frentes de Pareto, isto é, existem soluções em que os valores são diferentes para um dos objetivos, porém iguais para o outro objetivo em questão (COELLO; LAMONT; VELDHUIZEN, 2007).

⁴<https://products.office.com/pt-br/mac/microsoft-office-for-mac>

⁵<https://sourcegear.com/diffmerge/>

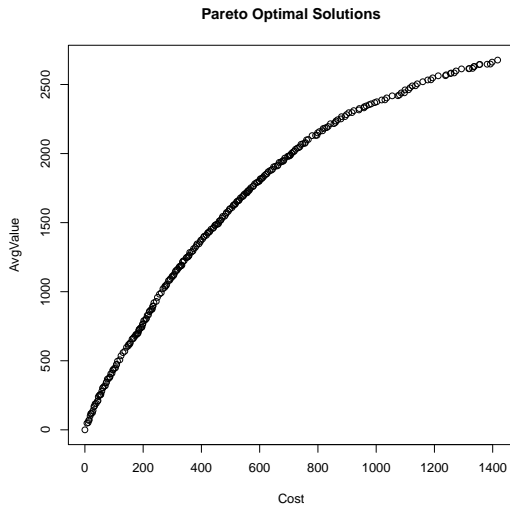


Figura 6.2: Frente de Pareto (dataset-1)

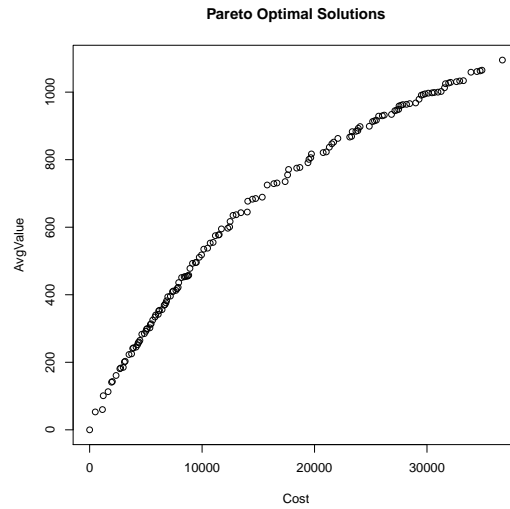


Figura 6.3: Frente de Pareto (dataset-2)

De forma oposta, Z3 produziu apenas soluções *strong Pareto*. Se eliminadas as soluções *weak Pareto*, os resultados apresentados seriam idênticos quanto ao número de soluções.

Em relação ao tempo necessário para a conclusão do processo de busca por soluções, os resultados indicam que o Yices exibe uma queda de performance dramática, necessitando de dias ou semanas para finalizar toda a busca (ver o caso da Tabela 6.3 para 50 requisitos). Comparativamente, o Z3 demonstra um desempenho muito melhor do que o Yices, principalmente, no que diz respeito à instância com maior número de requisitos. Devido a essas importantes características (menor tempo de execução e quantidade mais precisa de soluções), o Z3 foi selecionado como técnica SMT a ser aplicada no restante da abordagem proposta para o RA-MONRP.

Em seguida, três regiões de interesse foram identificadas manualmente em cada Frente de Pareto, representando valores pequenos, médios e grandes de orçamento (custo) disponível para a próxima versão do software. O Z3 foi então executado para obter as soluções em cada região de interesse, dentro das variações percentuais para custo (identificado como *cost*) e valor (identificado como *avgval*) estabelecidas no Tratamento 1 da primeira questão de pesquisa. Os resultados estão sumarizados nas Tabelas 6.5 e 6.6.

As Tabelas 6.5 e 6.6 mostram o custo e valor de cada solução identificada (pontos P1, P2 e P3) pertencentes às três regiões de interesse. Também é possível identificar as margens de tolerância para o custo e valor através da variação percentual (Δ) e o número de soluções encontradas próximas às soluções iniciais P1, P2 e P3 dentro da margem de tolerância estabelecida. Por fim, o tempo requerido para calcular e obter essas soluções também está reportado em cada tabela.

Para cada região de interesse em cada *dataset*, três diferentes visualizações envolvendo as soluções foram geradas:

- uma tabela e um histograma mostrando as triplas $\langle \text{custo}, \text{valor}, \text{risco} \rangle$ para cada solução;

Tabela 6.5: Resultados por região de interesse para o *dataset-1*

	cost	avgval	Região	Δ	Soluções	Tempo(s)
P1	202	790	ROI 1	5%	32	32,09
P2	958	2331	ROI 2	2%	53	22,39
P3	1335	2629	ROI 3	3%	40	0,45

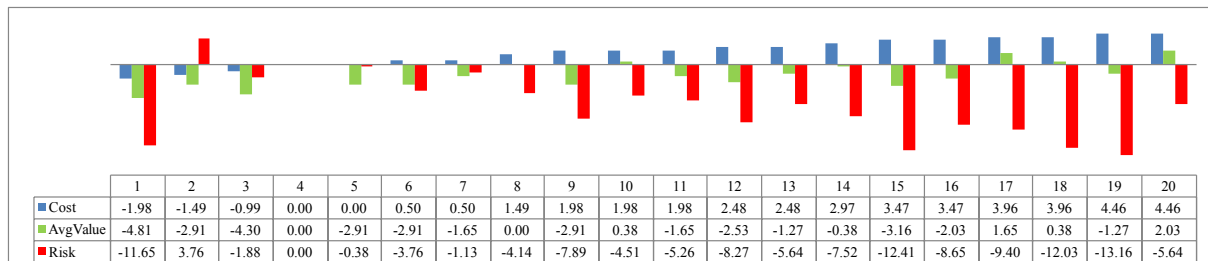
Tabela 6.6: Resultados por região de interesse para o *dataset-2*

	cost	avgval	Região	Δ	Soluções	Tempo(s)
P1	9150	493	ROI 1	5%	21	1,02
P2	18415	775	ROI 2	5%	42	8,74
P3	32910	1033	ROI 3	5%	36	0,20

- uma tabela e um histograma conjugado mostrando as variações de percentagem para custo, valor e risco para cada solução, comparada com o ponto P_i do ROI;
- um gráfico de tendência mostrando as variações de percentagem para as triplas $\langle \text{custo}, \text{valor}, \text{risco} \rangle$ através de segmentos conectados.

Para esse trabalho decidimos utilizar o histograma como ferramenta de visualização, pois nele é possível fazer a comparação das variações das percentagens de custo total, valor médio total e risco total entre as soluções na Região de interesse selecionada e o Ponto de referência escolhido. Desta forma, é possível se explorar as alternativas de variações de cenários e dar suporte à decisão na escolha de uma solução para o RAMONRP (PITANGUEIRA et al., 2016)

O gráfico de tendência, por outro lado, mostra claramente a propagação de perda/ganho de valores dos objetivos contra a redução/crescimento do risco. Para ilustrar a abordagem, considere a primeira região de interesse do *dataset-1* (ver Tabela 6.5). Na Figura 6.4 é possível ver uma amostra dos resultados para esta região de interesse e o gráfico de tendência para esses resultados pode ser visto na Figura 6.5 em que é possível se observar o comportamento conjugado entre as triplas custo, valor e risco.

Figura 6.4: Histograma de variações percentuais para a região de interesse 1 (*dataset-1*)

A Figura 6.4 mostra a percentagem de variação para 20 das 32 soluções ao redor da solução 4, que é a solução de referência (Ponto P1) escolhida na região de interesse 1. Por

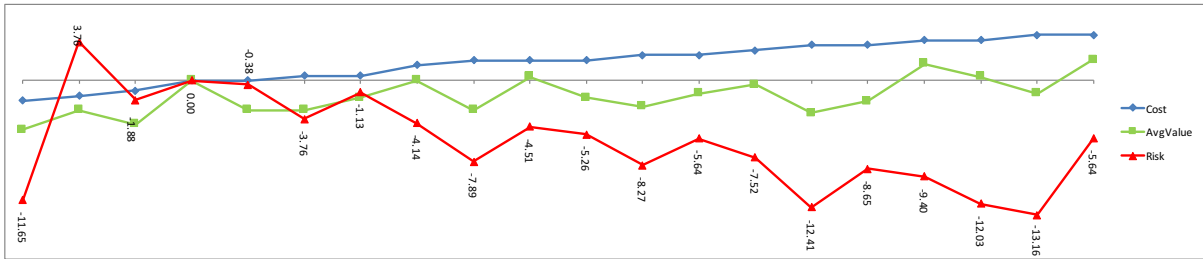


Figura 6.5: Tendência de variações para a região de interesse 1 (dataset-1)

essa solução ser a referência, ela não apresenta variações de percentagem para qualquer objetivo). As outras soluções encontradas na mesma região de interesse exibem alguma variação (positiva ou negativa) em cada objetivo. Uma variação negativa no risco indica que a solução apresenta um menor risco que a solução inicial do MONRP (solução 4). De forma similar, uma variação negativa (positiva) no custo ou valor indica menor (maior) custo ou valor na solução sendo analisada quando comparada à solução de referência.

Nesse histograma, a solução 19 apresenta a maior redução do risco (13,16%), conseguida porém com um aumento de custo de 4,46% e com uma perda de valor de 1,27%. Por outro lado, a solução 1 apresenta uma redução de risco de 11,65% associada a uma redução de custo de cerca de 1,98%, porém com perda de valor um pouco maior (4,81%).

A Figura 6.6 mostra a percentagem de variação para 30 das 53 soluções ao redor da solução 14, que é a solução de referência (Ponto P2) escolhida na região de interesse 2 para o dataset-1 e a Figura 6.7 apresenta o gráfico de tendência para esses resultados, onde é possível obter uma visualização geral do panorama de perda e ganho de cada objetivo.

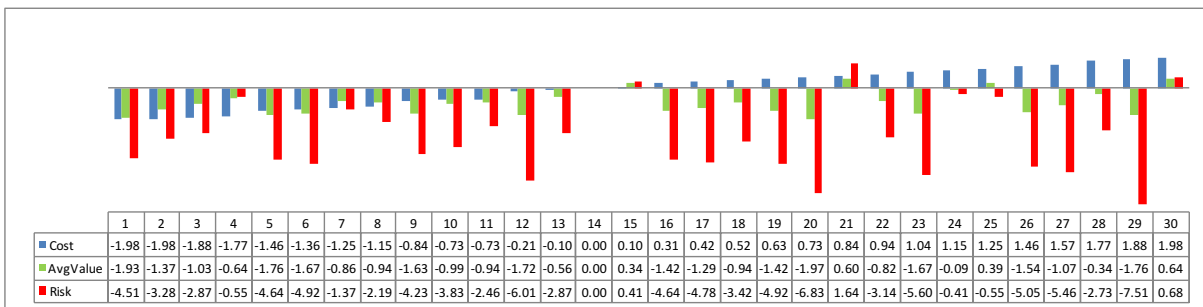


Figura 6.6: Histograma de variações percentuais para a região de interesse 2 (dataset-1)

Nesse histograma, a solução 29 apresenta a maior redução do risco (7,51%), conseguida com um acréscimo de custo de 1,88% e com uma perda de valor de 1,76%. Já a solução 12 mostra uma diminuição do risco em relação à solução original próximo a 6,01%, uma perda de valor por volta de 1,72% e, ao mesmo tempo, uma redução de custo em torno de 0,21%. Também no histograma é possível identificar a solução 20, que representa um aumento de 0,73% no custo, perda de 1,97% no valor e redução de 6,83% no risco.

Por fim, os resultados para a região de interesse 3 do dataset-1 podem ser vistos nas Figuras 6.8 e 6.9.

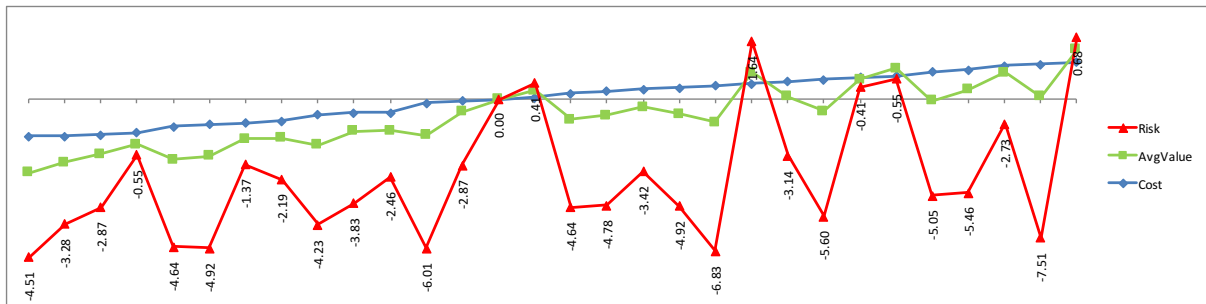


Figura 6.7: Tendência de variações para a região de interesse 2 (*dataset-1*)

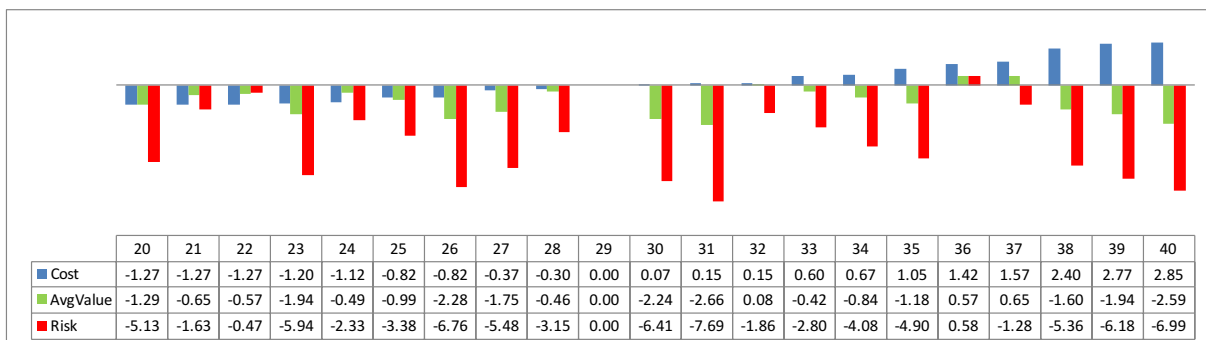


Figura 6.8: Histograma de variações percentuais para a região de interesse 3 (*dataset-1*)

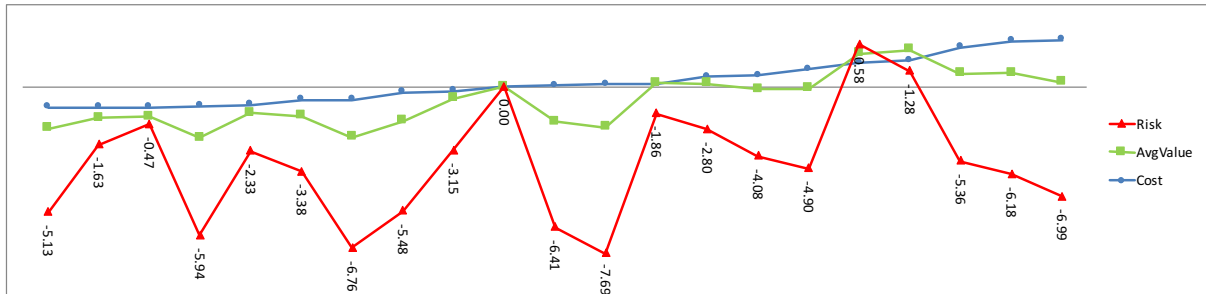


Figura 6.9: Tendência de variações para a região de interesse 3 (*dataset-1*)

Nesse cenário, 20 das 40 soluções obtidas são mostradas para fins de melhor visualização dos resultados. A solução 29 serve como ponto de referência para análise da vizinhança. A solução 31 apresenta a maior redução de risco (7,69%), conseguida à custa de apenas 0,15% de aumento de custo e uma redução de valor de 2,66%. Também pode ser identificada a solução 30, em que uma redução do risco na ordem de 6,41% é alcançada com um aumento de custo de apenas 0,07% e uma perda de valor de 2,24%. Para verificar o comportamento conjugado das soluções, a Figura 6.9 mostra as linhas de tendência para essa região de interesse.

No que diz respeito ao segundo conjunto de dados utilizados nos experimentos (*dataset-2*), os resultados a seguir mostram as soluções RA-MONRP para cada região de interesse. O histograma e o gráfico de tendências para a região de interesse do ponto P1 podem ser

vistos, respectivamente, nas Figuras 6.10 e 6.11.

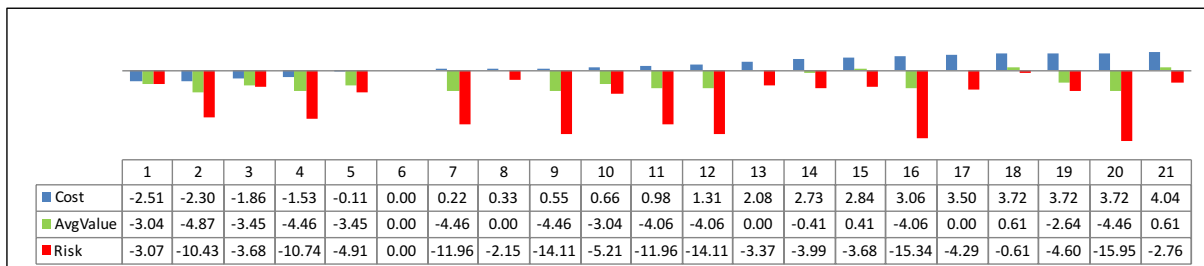


Figura 6.10: Histograma de variações percentuais para a região de interesse 1 (*dataset-2*)

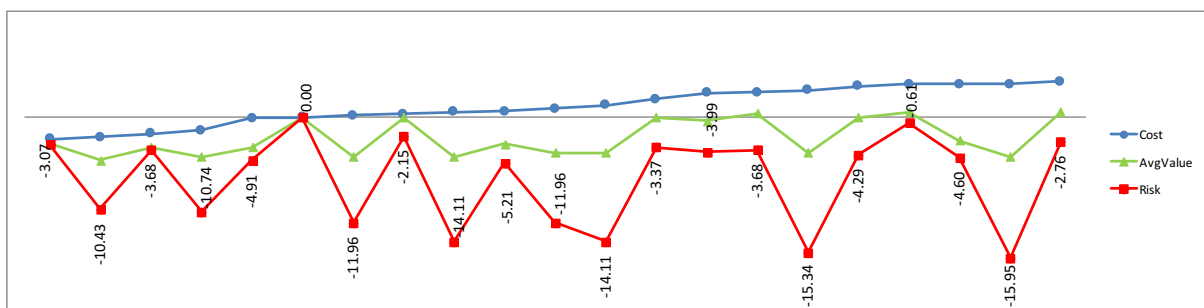


Figura 6.11: Tendência de variações para a região de interesse 1 (*dataset-2*)

A Figura 6.10 mostra todas as soluções obtidas para a região de interesse 1. Nesse caso, a solução 6 é tomada como ponto de referência para a análise da vizinhança e é possível perceber que a solução 20 possui a maior redução do risco (15,95%), acrescido de um aumento de custo de cerca de 3,72% e uma perda de valor próxima a 4,46%. Caso os *stakeholders* decidam por um menor aumento de custo e menor perda de valor, porém com uma redução de risco um pouco menor do que a observada na solução 20, a solução 12 poderá ser uma alternativa interessante, já que apresenta um acréscimo pequeno do custo (1,31%), uma perda de 4,06% no valor e uma redução de risco em torno de 14,11%. Outras soluções com boas reduções de risco podem ser citadas, tais como as soluções 4 e 7. O gráfico de tendências (Figura 6.11) mostra o comportamento das soluções e pode ser útil na identificação das soluções que apresentam maior redução de risco.

No que diz respeito à região de interesse 2 para o *dataset-2*, as Figuras 6.12 e 6.13 mostram os resultados obtidos.

Para essa região, 30 das 42 soluções encontradas são apresentadas. A solução 8 possui a maior redução de risco desse conjunto em relação à solução de referência (solução 11). Nesse caso, uma redução de risco próxima a 8,00% foi obtida com uma redução pequena de custo (em torno de 0,92%), porém com uma perda de valor da ordem de 4,90%.

Para finalizar a análise do Tratamento 1, as Figuras 6.14 e 6.15 mostram os resultados dos dados para a terceira região de interesse do *dataset-2*.

No histograma, para fins de melhor visualização, são mostradas as percentagens de variação para 21 das 36 soluções ao redor da solução 23, que é a solução MONRP de

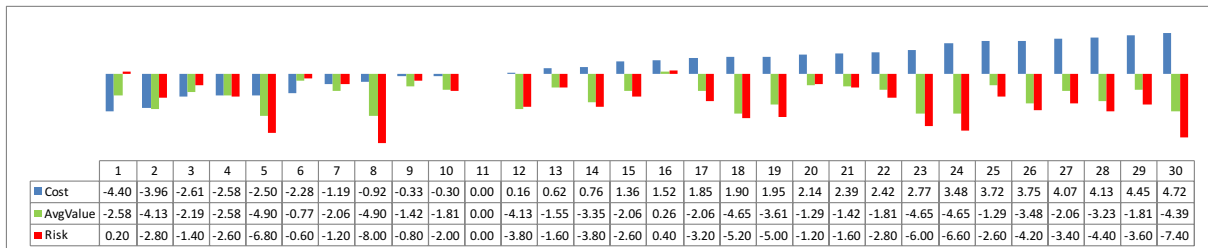


Figura 6.12: Histograma de variações percentuais para a região de interesse 2 (*dataset-2*)

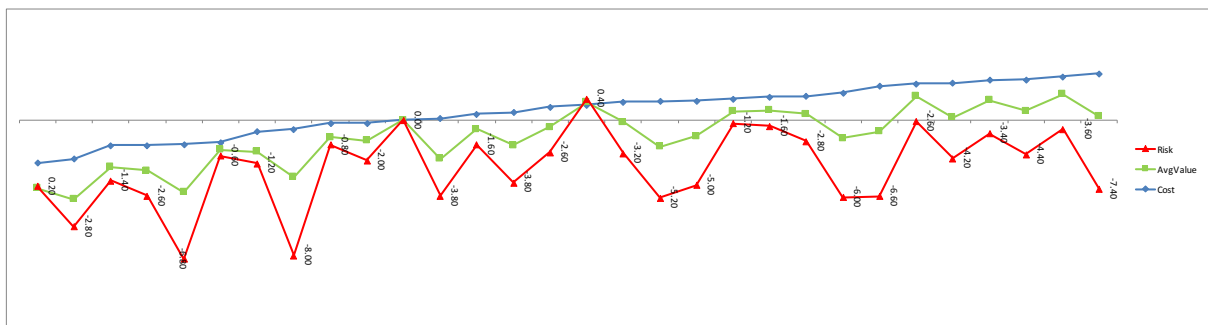


Figura 6.13: Tendência de variações para a região de interesse 2 (*dataset-2*)

referência escolhida para esta região de interesse. Nessa amostra, é possível identificar que a solução 24 apresenta a maior redução do risco (6,38%) em relação à solução de referências. Esta redução foi obtida com aumento de custo pequeno (0,43%) e com uma redução de valor não tão insignificante (4,84%). Também para esse conjunto, a solução 28 apresenta uma redução de risco de 6,23%, mas com um aumento de custo de 2,02% e perda de valor próxima a 4,74%. Essas duas soluções podem ser vistas como os maiores picos de redução de risco no gráfico de tendências apresentado na Figura 6.15.

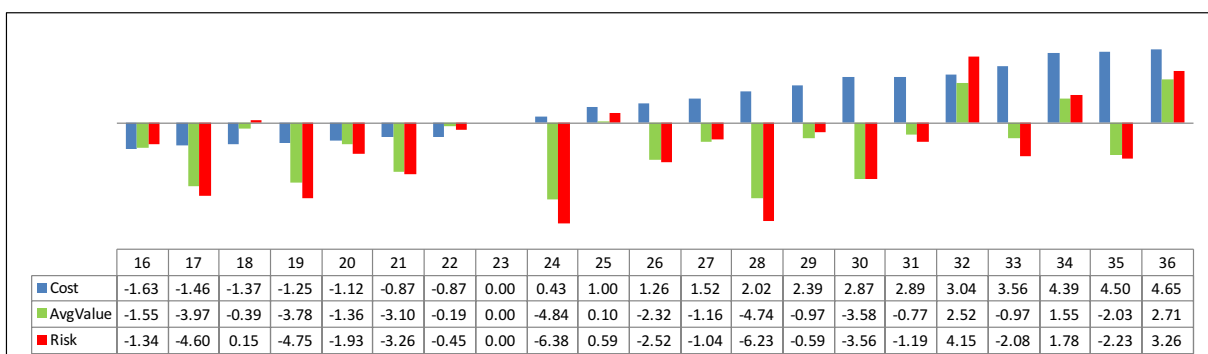


Figura 6.14: Histograma de variações percentuais para a região de interesse 3 (*dataset-2*)

6.3.1.2 Tratamento 2 No Tratamento 2 novos cenários foram adotados com o objetivo de explorar o espaço de soluções. Para tanto, um dos objetivos possui uma margem de tolerância fixa e o outro objetivo varia livremente. Nesse cenário, os mesmos pon-

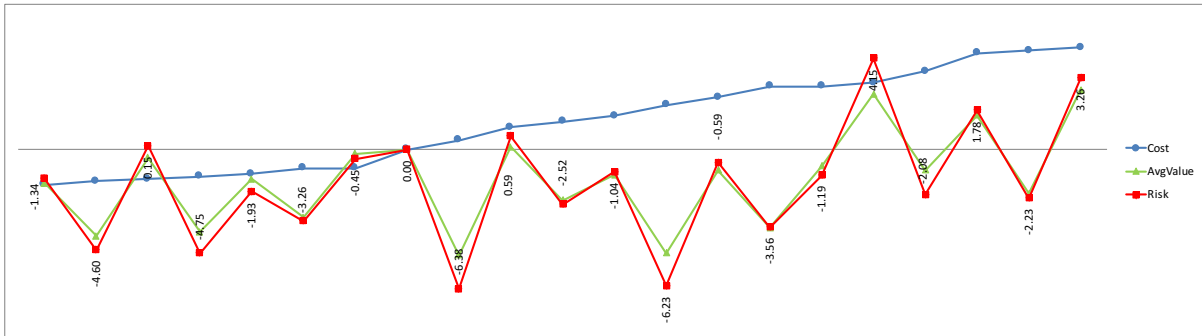


Figura 6.15: Tendência de variações para a região de interesse 3 (*dataset-2*)

tos centrais de cada região de interesse previamente utilizados no Tratamento 1 foram utilizados no Tratamento 2 (ver Tabelas 6.5 e 6.6).

As Tabelas 6.7 e 6.8 apresentam os resultados obtidos nos cenários analisados neste experimento. Por exemplo, considere o *dataset-1* (Tabela 6.7) e a região de interesse 3, que possui uma margem de tolerância de 3%. Duas situações podem ser identificadas: a primeira, quando o valor varia livremente (*Free AvgValue*) e o custo obedece a variação especificada de 3% (Δ) e a segunda quando a variação do custo não possui nenhuma limitação (*Free Cost*) e o valor varia até 3%.

Tabela 6.7: Dataset-1: Resultados para Tratamento 2

Dataset 1			
		Free AvgValue	Free Cost
ROI 1 ($\Delta = 5\%$)	tempo(s)	1.305,16	timeout
	soluções	217	279
ROI 2 ($\Delta = 2\%$)	tempo(s)	timeout	4.394,09
	soluções	30	439
ROI 3 ($\Delta = 3\%$)	tempo(s)	642,94	0,84
	soluções	206	67
Total de soluções		453	785

Como pode ser identificado nas tabelas 6.7 e 6.8, os resultados de tempo de execução e número de soluções na Frente de Pareto apresentam diferenças expressivas se comparados aos resultados do Tratamento 1 (Tabelas 6.5 e 6.6). Esses resultados, em especial o tempo de execução, são proporcionais ao aumento do espaço de busca que é observado quando um dos objetivos não está restrito. Assim, em um espaço de busca maior, há maiores possibilidades de novas soluções serem consideradas Pareto e, conseqüentemente, o tempo de execução para o processo de otimização do RA-MONRP também aumenta. Importante ressaltar que para o *dataset-1*, que possui o dobro de requisitos de software do *dataset-2*, o tempo de execução para a busca extrapolou o tempo máximo estabelecido (um dia) em duas situações. Esse tempo limite (*timeout*) aconteceu provavelmente devido

Tabela 6.8: Dataset-2: Resultados para o Tratamento 2

Dataset 2			
		Free AvgValue	Free Cost
ROI 1 ($\Delta = 5\%$)	tempo(s)	156,29	1.521,35
	soluções	235	199
ROI 2 ($\Delta = 5\%$)	tempo(s)	12,37	554,61
	soluções	482	375
ROI 3 ($\Delta = 5\%$)	tempo(s)	1,95	0,35
	soluções	97	53
Total de soluções		814	627

à necessidade de se excluir soluções UNSAT e por computar um número maior de soluções SAT que não são ótimas.

A fim de evitar a repetição do padrão de análise e manter o foco nos aspectos essenciais da redução do risco de insatisfação dos *stakeholders*, apenas um cenário para cada *dataset* (Tabelas 6.7 e 6.8) foi escolhido para ser discutido a seguir. Os demais cenários podem ser vistos no apêndice B.

A Figura 6.16 mostra o histograma e a Figura 6.17 mostra o gráfico de tendências para a região de interesse 1 do *dataset-1* para o cenário em que o valor varia livremente e o custo varia até 5%. Uma amostra de 29 das 217 soluções são apresentadas nestes gráficos.

Nesse caso, é possível identificar soluções que possuem um decréscimo acentuado do risco. Por exemplo, a solução 7 apresenta uma redução de, aproximadamente, 18,05% do risco, acompanhada de uma redução do custo de 3,47% e uma perda de valor considerável em torno de 9,49%. Outras soluções apresentam uma redução de risco acima de 10%, como as soluções 1, 2 e 4. No entanto, essa redução também é acompanhada com perdas de valor acima de 7,5%.

Em outros casos, existem soluções que apresentam boa redução de risco com pequena perda de valor. No entanto, há um custo a ser pago para tal combinação. Por exemplo,

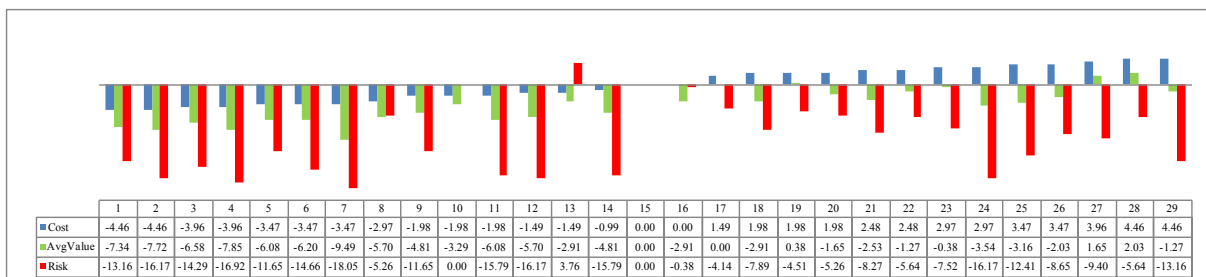


Figura 6.16: Histograma de variações percentuais para a região de interesse 1 (*dataset-1*) com valor livre

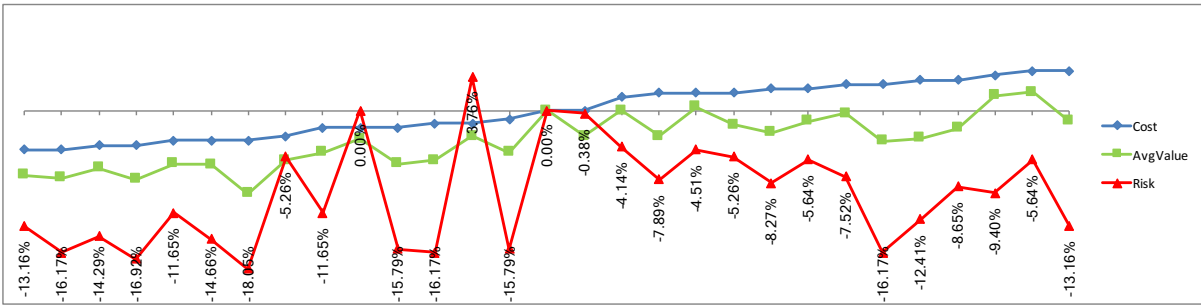


Figura 6.17: Tendência de variações para a região de interesse 1 (*dataset-1*) com valor livre

a solução 29 mostra uma redução de 13,16% no risco, uma perda de valor de apenas 1,27%, porém com um aumento do custo por volta de 4,46%. Exemplos desse mesmo comportamento podem ser identificados nas soluções 24 e 25. O gráfico de tendências da Figura 6.17 mostra o comportamento conjunto das soluções para o RA-MONRP, sendo possível identificar os picos para cada um dos três objetivos.

Para o *dataset-2* (ver Tabela 6.8), selecionamos o cenário da região de interesse 1 com custo livre e valor fixo para apresentação dos resultados. Os gráficos com resultados observados nestes são apresentados nas Figuras 6.18 e 6.19.

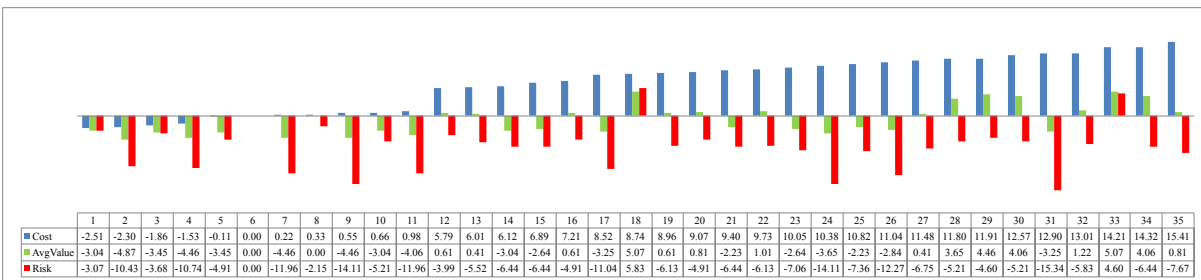


Figura 6.18: Histograma de variações percentuais para a região de interesse 1 (*dataset-2*), com Custo Livre

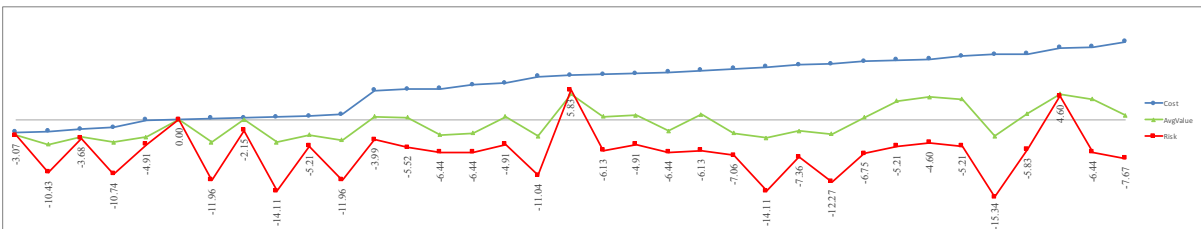


Figura 6.19: Tendência de variações para a região de interesse 1 (*dataset-2*), com custo livre

Para esse cenário, uma amostra de 35 soluções são apresentadas dentre as 199 encontradas pelo processo de busca. Nesse caso, a solução 9 apresenta uma variação negativa

acentuada para o risco em relação à solução de referência (solução 6). No caso, a redução do risco aproximada é de 14,11%, com uma redução do valor em cerca de 4,46% e um aumento do custo pequeno, próximo de 0,55%.

Outras soluções interessantes também podem ser ressaltadas. Por exemplo, a solução 7 apresenta uma redução para o risco em torno de 11,96%, com uma redução do valor em cerca de 4,46% e um aumento do custo pequeno (0,22%). Já a solução 2 difere das soluções citadas anteriormente porque apresenta uma redução no custo (ao invés de acréscimo) em torno de 2,30%, com uma perda de valor de 4,87% e uma redução de risco (10,43%) não tão menor do que aquela apresentada pelas soluções vizinhas 7 e 9.

A solução 31 possui uma das maiores reduções do risco (15,34%). No entanto, seu custo aumenta consideravelmente, sendo uma solução 12,90% mais cara do que a solução de referência. No gráfico de tendências é possível identificar as soluções que possuem os maiores picos de redução de risco e avaliar a combinação entre perda e ganho para os demais objetivos.

6.3.1.3 Tratamento 3 O Tratamento 3 é uma variante do tratamento 1 em que os objetivos possuem margens de tolerância fixa, podendo ser um valor de margem para um objetivo e um valor distinto para o segundo objetivo do MONRP.

Os mesmos pontos centrais de cada região de interesse previamente utilizados no Tratamento 1 foram utilizados no Tratamento 3 (ver Tabelas 6.5 e 6.6). A expansão de cada margem de tolerância foi obtida escolhendo uma nova margem, sendo o limite mínimo inferior o valor do Δ no Tratamento 1 e o limite superior máximo sendo o dobro da margem mais alta no conjunto de dados original. Por exemplo, quando a margem de tolerância original for de 5%, esse valor será o limite mínimo e o limite máximo será 10%. As novas margens foram obtidas usando uma função para geração de números aleatórios e obedecendo os critérios estabelecidos para limites mínimo e máximo. Os resultados para ambos datasets podem ser vistos nas Tabelas 6.9 e 6.10.

Tabela 6.9: Resultados por região de interesse para *dataset-1* (Tratamento 3)

	cost	avgval	Region	Δ cost	Δ avgvalue	Solutions	Time(s)
P1	202	790	ROI 1	6%	9%	65	75,46
P2	958	2331	ROI 2	4%	6%	256	4.204,86
P3	1335	2629	ROI 3	9%	10%	230	169,90

Tabela 6.10: Resultados por região de interesse para *dataset-2* (Tratamento 3)

	cost	avgval	Region	Δ cost	Δ avgvalue	Solutions	Time(s)
P1	9150	493	ROI 1	10%	7%	58	3,52
P2	18415	775	ROI 2	7%	8%	138	78,25
P3	32910	1033	ROI 3	6%	10%	82	1,16

É possível perceber que o número de soluções e o tempo de execução exigido para

completar a busca são maiores para todos os cenários quando comparados aos dados obtidos após a execução do Tratamento 1. Possivelmente, deve-se o aumento de grandeza desses valores ao aumento do espaço de busca.

Da mesma forma que o Tratamento 2, a fim de evitar a repetição do padrão de análise e manter o foco nos aspectos essenciais da redução do risco de insatisfação dos *stakeholders*, apenas um cenário para cada *dataset* foi selecionado e será discutido nos próximos parágrafos. Os demais cenários podem ser vistos no apêndice C.

O histograma da Figura 6.20 mostra os resultados para o *dataset-1*, na região de interesse 2. Uma amostra de 30 das 256 soluções são apresentadas. Algumas soluções apresentaram boa redução do risco. Por exemplo, a solução 13 mostra uma diminuição do risco na ordem de 13,52%, acompanhada de uma diminuição do custo em cerca de 0,21%; porém, esta solução apresenta uma perda de valor de 5,02%. De forma similar, a solução 6 apresenta tanto redução de risco quanto de custo em 12,70% e 2,71%, respectivamente, e perda do valor um pouco maior, por volta de 5,32%.



Figura 6.20: Histograma de variações percentuais para a região de interesse 2 (*dataset-1*), Tratamento 3

No geral, é possível identificar que nessa composição a maioria das soluções que apresentam maiores reduções de risco também apresentam uma perda considerável de valor. Esse comportamento pode ser visualizado de forma conjugada para os objetivos no gráfico de tendência exposto na Figura 6.21.

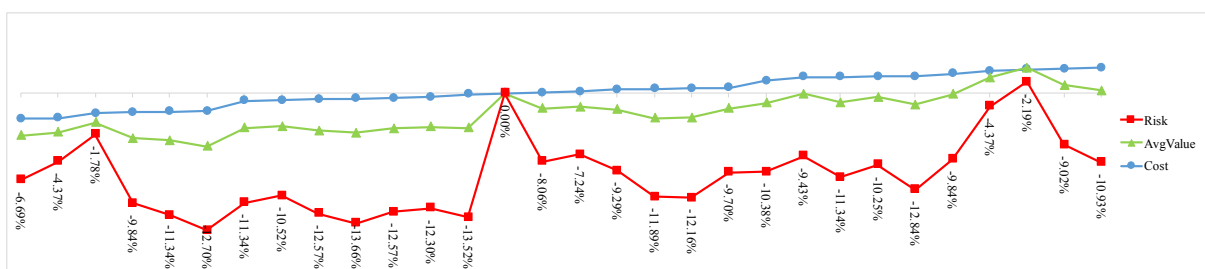


Figura 6.21: Tendência de variações para a região de interesse 2 (*dataset-1*), Tratamento 3

Em relação aos resultados para o *dataset-2*, as Figuras 6.22 e 6.23 mostram os resultados para a região de interesse 2, no caso do Tratamento 3. Bons índices de redução do risco foram obtidos na vizinhança da solução de referência (solução 12). Por exemplo, a solução 27 apresenta uma redução do risco em cerca de 12,40%, porém com aumento do custo de 6,05% e uma perda de valor significativa, próxima a 7,74%.

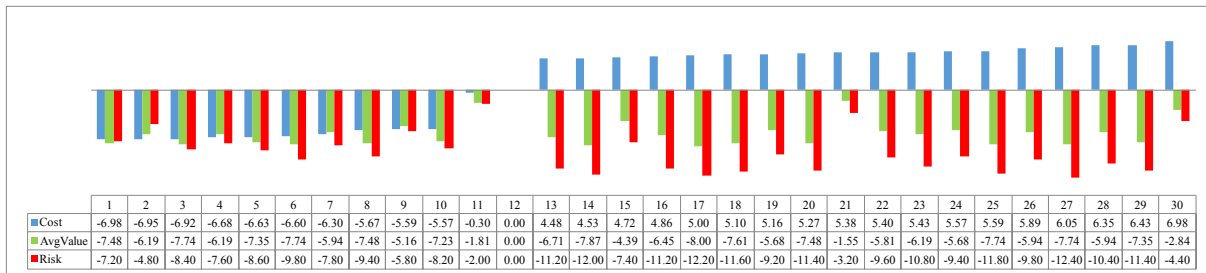


Figura 6.22: Histogramas de variações de percentuais para a região de interesse 2 (*dataset-2*), Tratamento 3

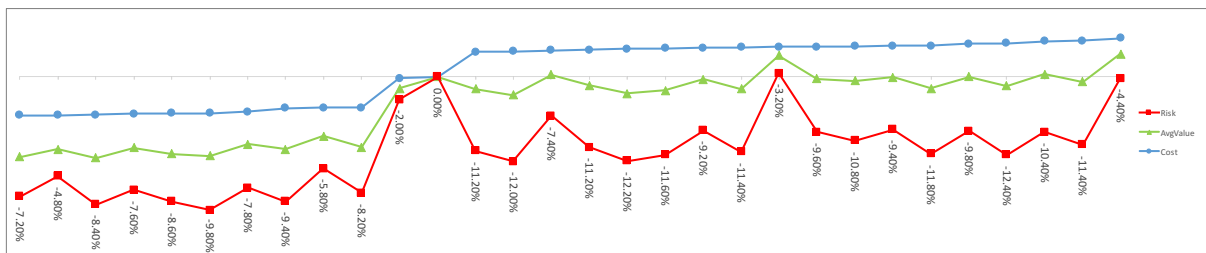


Figura 6.23: Tendência de variações para a região de interesse 2 (*dataset-2*), Tratamento 3

De forma similar, a solução 17 apresenta uma redução do risco em torno de 12,20%, com uma perda também acentuada de valor (próximo a 8,00%) e com um aumento do custo em torno de 5%. A solução 29 também é um exemplo de boa redução de risco (11,40%) conjugada com aumento de custo (6,43%) e perda de valor (7,35%) acentuados.

Porém, nesse cenário é possível observar soluções com redução de risco e redução de custo conjugadas. Por exemplo, a solução 7 apresenta uma redução de custo de 6,30%, uma redução do risco de 7,80% e uma perda de valor em torno de 5,94%. Outras soluções apresentam configuração similar à solução 7, tais como as soluções de 1 a 11.

6.3.2 Resultados para Teste de Escalabilidade das Técnicas - RQ2

Essa seção apresenta os resultados do teste de escalabilidade para três situações distintas: (1) aumento conjunto do número de requisitos e dependências; (2) crescimento apenas do número de requisitos; e (3) aumento apenas do número de dependências. Os resultados obtidos para a escalabilidade e apresentados nessa seção seguem o procedimento descrito na seção 6.2.2 para a expansão e geração de novas instâncias.

Todas as análises foram realizadas para os algoritmos Z3 e NSGA-II e apenas para o *dataset-2*, pois a execução da instância original do *dataset-1* no MONRP (50 requisitos) já apresentou uma diferença de tempo de execução muito expressiva entre os dois algoritmos. Para o *dataset-1*, o Z3 percorria o espaço de busca em 183.320,29 segundos e gerava uma Frente de Pareto com 285 soluções. Neste mesmo cenário, o NSGA-II realizava o percurso em um tempo médio de aproximadamente 50,25 segundos, encontrando 351 soluções. Portanto, desde a configuração original a meta-heurística superava o tempo de execução

da técnica determinística SMT por um fator maior do que 3.000 vezes.

6.3.2.1 Escalabilidade para aumento conjunto de requisitos e dependências

A Tabela 6.11 mostra os resultados obtidos para o *dataset-2*. O número de requisitos foi aumentado em duas unidades a cada instância do experimento. A tabela mostra o tempo de execução e o número de soluções encontradas pelo SMT Z3 e pela meta-heurística NSGA-II. A meta-heurística foi executada 100 vezes para cada instância e os resultados mostram o tempo médio e a média do número de soluções computadas ao longo das 100 execuções.

Tabela 6.11: Resultados para escalabilidade - aumento conjunto requisitos-dependências

Reqs	Deps	Tempo-Z3	Tempo-NSGA-II	Soluções-Z3	Soluções-NSGA-II
25	11	81,86	21,95	143	170
27	12	273,19	24,26	165	186
29	13	960,83	28,59	213	221
31	14	3.410,90	29,89	230	238
33	15	8.342,45	32,50	243	255
35	16	36.175,45	37,85	259	268
37	17	77.336,76	46,61	284	295
39	18	203.239,20	49,68	302	312

Para o *dataset-2*, o fator médio (*average_factor*) entre requisitos e dependências foi calculado como 0,44 (equação 6.4). Portanto, no caso da inclusão de novas dependências, sempre que *inc_deps* ficou abaixo de um número inteiro, esse valor foi aproximado para uma unidade. O exemplo abaixo mostra o cálculo para a instância original do *dataset-2*.

$$average_factor = |Deps| / |Reqs| = 0,44 \quad (6.4)$$

$$inc_deps = |NewReqs| * |average_factor| = 2 * 0,44 = 0,88 \quad (6.5)$$

Os resultados da Tabela 6.11 mostram que o NSGA-II é substancialmente mais rápido que o Z3, especialmente quando a instância do problema se torna maior. Pode ser notado que a cada nova instância o desempenho em tempo do Z3 piora de forma bastante acentuada. De forma oposta, o desempenho em tempo do NSGA-II apresenta apenas um ligeira queda. Em relação ao número de soluções obtidas para cada instância, por ser uma técnica determinística, o Z3 sempre encontra o mesmo conjunto de soluções por instância. Por outro lado, o NSGA-II é uma técnica probabilística e calcula uma aproximação da Frente de Pareto a cada nova execução do algoritmo. O ponto de parada para execução desse experimento foi definido quando o Z3 alcançou uma grandeza de dias para percorrer o espaço de busca.

Tabela 6.12: Resultados para escalabilidade aumentando apenas o número de requisitos

Reqs	Deps	Tempo-Z3	Tempo-NSGA-II	Soluções-Z3	Soluções-NSGA-II
25	11	81,86	21,95	143	170
27	11	418,38	23,60	171	193
29	11	1.952,61	32,38	222	252
31	11	8.047,95	34,39	251	282
33	11	30.136,87	42,00	271	304
35	11	168.452,52	46,72	293	324

6.3.2.2 Escalabilidade para aumento apenas do número de requisitos Essa seção apresenta os resultados obtidos quando o número de requisitos é aumentado, enquanto o número de dependências permanece o mesmo. Para esse cenário, o número de requisitos foi aumentado em duas unidades ($r = 2$) a cada nova instância gerada. A Tabela 6.12 mostra os resultados obtidos.

Nesse cenário, o tempo de execução do Z3 cresce exponencialmente de uma execução para a próxima. Por exemplo, com 35 requisitos, o Z3 precisava de mais de um dia para produzir a frente de Pareto. Em comparação com os resultados anteriores (aumentando tanto requisitos quanto dependências - Tabela 6.11), o SMT exibiu um tempo de execução semelhante quando a instância variante do *dataset-2* tinha 39 requisitos. Pode se conjecturar que as restrições impostas pelas dependências são usadas pelo Z3 para limitar o espaço de busca. De fato, quando novos requisitos são adicionados sem aumentar proporcionalmente o número de restrições, o Z3 tem de explorar um espaço maior de pesquisa, exigindo mais de tempo de computação.

No geral, o algoritmo NSGA-II apresentou um desempenho muito melhor que o Z3. Além disso, comparando a Tabela 6.12 com a Tabela 6.11, o desempenho da NSGA-II continua a ser bastante estável, com apenas alguns segundos de diferença a medida que as instâncias maiores lhe são apresentadas. No que diz respeito ao número de soluções, tal como descrito anteriormente, o NSGA-II produziu mais soluções do que Z3. Na terceira questão de pesquisa analisaremos a qualidade destas soluções.

6.3.2.3 Escalabilidade para aumento apenas do número de dependências Neste experimento, o número de requisitos é mantido inalterado, enquanto o número de dependências é aumentado em uma unidade por vez. A Tabela 6.13 apresenta os resultados encontrados pela execução do experimento.

É importante observar que, com mais de 18 dependências, o SMT Z3 torna-se mais rápido do que a metaheurística NSGA-II. Pode se conjecturar que a computação de soluções em um espaço de busca severamente limitado é mais fácil para Z3 do que em um espaço de busca com restrições mais leves, a ponto de superar o desempenho (tempo de execução) do NSGA-II quando o número de restrições se torna grande o suficiente.

No que diz respeito às soluções obtidas, como esperado o NSGA-II produz mais soluções do que o Z3, mas, diferentemente dos experimentos anteriores, diminui o número de soluções a cada instância sucessiva. Isso acontece porque o espaço é cada vez mais

Tabela 6.13: Resultados para escalabilidade aumentando o número de dependências

Reqs	Deps	Tempo-Z3	Tempo-NSGA-II	Soluções- Z3	Soluções-NSGA-II
25	11	81,86	21,95	143	170
25	12	48,58	22,49	137	160
25	13	39,58	19,73	132	145
25	14	33,81	20,30	126	152
25	15	27,67	20,91	126	162
25	16	26,49	18,86	127	144
25	17	26,73	21,06	141	156
25	18	22,67	18,68	118	143
25	19	15,73	18,50	110	141
25	20	10,66	18,57	110	142
25	21	7,88	18,72	111	142
25	22	3,34	18,86	109	143
25	23	2,96	18,77	111	145
25	24	2,15	18,09	99	139
25	25	1,96	18,06	98	134

restrito dado que o número de dependências é aumentado.

6.3.3 Resultados para Comparação da Qualidade das Frentes de Pareto - RQ3

Nessa seção são apresentados os testes comparativos entre as Frentes de Pareto obtidas pela meta-heurística NSGA-II e pela técnica determinística Z3. Esta comparação é importante porque os resultados das Frentes de Pareto podem apresentar dissimilaridades quando técnicas distintas são usadas, porque as estratégias de execução da busca podem ser diferentes e uma estratégia pode ser mais eficiente que a outra. Para obtenção e comparação dos resultados, o protocolo de execução dos experimentos obedeceu a abordagem descrita na seção 6.2.3.

Foram utilizados dois panoramas de comparação. O primeiro deles, apresentado na subseção 6.3.3.1, aborda a diferença percentual entre as soluções que o NSGA-II encontrou, mas que não estavam presentes na técnica determinística. Ou seja, são confrontadas as soluções sub-ótimas encontradas na busca exercida pelo NSGA-II em comparação com a Frente de Pareto ótima obtida pelo Z3. O segundo panorama diz respeito à comparação de soluções ótimas, ou seja, quantas soluções exatamente iguais à técnica determinística Z3 o NSGA-II encontrou. Nesse caso, o oposto do descrito acima é realizado, pois as Frentes de Pareto são confrontadas e comparadas para se verificar a quantidade de soluções iguais em ambas as técnicas.

6.3.3.1 Resultados para diferença percentual média entre as Frentes de Pareto obtidas pelas técnicas de busca A Tabela 6.14 mostra o conjunto de dados utilizado, a quantidade de soluções obtidas pela técnica determinística, a quantidade de soluções obtidas pela técnica probabilística, a diferença média (Avg $\Delta\%$) de valor/custo e

a máxima diferença percentual ($\text{Max } \Delta \%$) para custo/valor entre as soluções produzidas pelo Z3 e as soluções sub-ótimas produzidas pelo NSGA-II.

Tal como descrito na Seção 6.2.3, a diferença média foi calculada comparando-se a solução obtida pelo NSGA-II (custo, valor) com a solução com custo/valor mais próximo na Frente de Pareto obtida pelo Z3. Dentre as diferenças calculadas, é possível identificar a maior diferença (ou máxima diferença percentual) na comparação entre as técnicas de busca. Podemos notar que, em média, tais diferenças são pequenas, embora ocasionalmente a meta-heurística NSGA-II possa produzir soluções que diferem por um percentual não negligenciável da Frente de Pareto encontrada pela técnica determinística (Z3).

Tabela 6.14: Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg $\Delta \%$ (cost)	Max $\Delta \%$ (cost)	Avg $\Delta \%$ (value)	Max $\Delta \%$ (value)
dataset-1	285	351	0,3979%	1,2658%	0,2664%	1,8145%
dataset-2	143	170	0,3826%	0,8241%	0,5740%	2,3448%

Mesmo levando em consideração a maior diferença percentual encontrada tanto para o custo quanto para o valor, a grandeza numérica permite afirmar que há uma aproximação razoável entre as Frentes de Pareto de ambas as técnicas. As Tabelas 6.15, 6.16 e 6.17 mostram os resultados do teste de aproximação das Frentes de Pareto para os resultados dos testes de escalabilidade realizados para o *dataset-2*.

Mais uma vez, pode-se notar que, em média, essas diferenças são muito pequenas tanto para o *dataset-1* quanto para o *dataset-2* (ver Tabela 6.14), porque ocasionalmente o NSGA-II produz soluções que, embora sejam sub-ótimas, são próximas das soluções obtidas pelo Z3. No caso do *dataset-1*, a máxima diferença percentual foi encontrada quanto o objetivo era o custo, enquanto para o *dataset-2* a máxima diferença foi para o valor médio dos requisitos.

Os resultados mostram que aumentando apenas as dependências (Tabela 6.17), a diferença percentual tende a ser maior do que os resultados obtidos nos outros dois casos (aumento de requisitos e requisitos em conjunto com dependências).

Tabela 6.15: Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao crescimento do número de requisitos e dependências (REQ/DEP)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg $\Delta \%$ (cost)	Max $\Delta \%$ (cost)	Avg $\Delta \%$ (value)	Max $\Delta \%$ (value)
dataset-2	143	170	0,3826%	0,8241%	0,5740%	2,3448%
(27/12)	165	186	0,3739%	0,8459%	0,2695%	0,7112%
(29/13)	213	221	0,3389%	0,7951%	0,2472%	0,6485%
(31/14)	230	238	0,2964%	0,6575%	0,2794%	0,6353%
(33/15)	243	255	0,2934%	0,5964%	0,3224%	0,6543%
(35/16)	259	268	0,3338%	0,5476%	0,2772%	0,6269%
(37/17)	284	295	0,3350%	0,5720%	0,2673%	0,5681%
(39/18)	302	312	0,3083%	0,6234%	0,2361%	0,5479%

Tabela 6.16: Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao crescimento do número de requisitos (REQ)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg $\Delta\%$ (cost)	Max $\Delta\%$ (cost)	Avg $\Delta\%$ (value)	Max $\Delta\%$ (value)
dataset-2	143	170	0,3826%	0,8241%	0,5740%	2,3448%
(27)	171	193	0,3629%	0,8491%	0,2271%	0,7112%
(29)	222	252	0,3171%	0,8753%	0,1936%	0,6527%
(31)	251	282	0,2652%	0,7254%	0,2801%	2,0814%
(33)	271	304	0,1892%	0,5059%	0,1664%	0,5555%
(35)	293	324	0,1953%	0,4345%	0,1975%	0,9353%

Tabela 6.17: Resultados para teste de escalabilidade - Diferença percentual média entre as soluções determinísticas (Z3) e probabilísticas (NSGA-II) quanto ao aumento do número de dependências (DEP)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg $\Delta\%$ (cost)	Max $\Delta\%$ (cost)	Avg $\Delta\%$ (value)	Max $\Delta\%$ (value)
dataset-2	143	170	0,3826%	0,8241%	0,5740%	2,3448%
(12)	137	160	0,3827%	0,8241%	0,5740%	2,3448%
(13)	132	145	0,3826%	0,8241%	0,5740%	2,3448%
(14)	126	152	1,0871%	2,5784%	0,6356%	2,3448%
(15)	126	162	1,1025%	2,5784%	0,7855%	2,7692%
(16)	127	144	0,6169%	2,3679%	0,4927%	2,3448%
(17)	141	156	0,6266%	1,8821%	0,3459%	0,6787%
(18)	118	143	0,6969%	2,3023%	0,5515%	2,2457%
(19)	110	141	0,7203%	2,6995%	0,6603%	3,9851%
(20)	110	142	0,7203%	2,6995%	0,6447%	3,9850%
(21)	111	142	0,7274%	2,6995%	0,6702%	3,9850%
(22)	109	143	0,7581%	2,6995%	0,6754%	3,5754%
(23)	111	145	0,7543%	2,6995%	0,7748%	3,7735%
(24)	99	133	0,7369%	2,6995%	0,7168%	3,5754%
(25)	98	134	0,9406%	3,8323%	0,6746%	3,1638%

6.3.3.2 Resultados para teste de aproximação das Frentes de Pareto Os resultados apresentados nas Tabelas 6.18 a 6.21 mostram o quanto a técnica probabilística (NSGA-II) se aproxima da técnica determinística em termos de igualdade de soluções, ou seja, quantas soluções iguais ao Z3 foram encontradas pela técnica probabilística. Como explicado anteriormente, o NSGA-II, por ser uma técnica que se baseia em números aleatórios para exercer seu processo de busca, pode incluir no conjunto de soluções tanto soluções iguais ou próximas do que poderia ser a Frente de Pareto obtida por uma técnica determinística. Isso pode ser percebido nas diversas execuções do NSGA-II, comparando-se o número de soluções obtidas por essa técnica com os resultados do SMT Z3.

A Tabela 6.18 mostra os resultados para os conjuntos de dados originais. São apresentados a quantidade de soluções obtidas pelo Z3, a quantidade de soluções obtidas pelo NSGA-II, a média de soluções iguais obtidas pelo NSGA-II e o percentual médio de soluções iguais em relação aos resultados do Z3.

No caso dos resultados originais, é possível perceber que a eficácia do NSGA-II para o *dataset-2* é muito maior do que para o *dataset-1*. Mas é importante ressaltar que o espaço de busca para o *dataset-2* (25 requisitos) é muito menor do que o espaço de busca para o primeiro (50 requisitos), o que pode diminuir a eficácia no processo de otimização,

Tabela 6.18: Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 para os conjuntos de dados originais

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg. Soluções Iguais	Avg aproximação %
dataset-1	285	351	122	42,80%
dataset-2	143	170	134	93,70%

que pode ser afetado por múltiplos conflitos e decisões baseados na probabilidade.

Os próximos resultados estão relacionados com os dados obtidos através dos experimentos conduzidos para avaliar a escalabilidade das técnicas no *dataset-2*. A Tabela 6.19 e a Figura 6.24 mostram os resultados quanto ao crescimento do número de requisitos e dependências. É possível notar que o NSGA-II obteve boas aproximações em relação à técnica determinística, mesmo considerando que seus resultados incluem algumas soluções sub-ótimas.

Tabela 6.19: Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao crescimento do número de requisitos e dependências (REQ/DEP)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg. Soluções Iguais	Avg aproximação %
dataset-2	143	170	134	93,70%
(27/12)	165	186	155	93,93%
(29/13)	213	221	203	95,30%
(31/14)	230	238	219	95,21%
(33/15)	243	255	234	96,29%
(35/16)	259	268	247	95,36%
(37/17)	284	295	272	95,77%
(39/18)	302	312	290	96,02%

No gráfico de tendência apresentado na Figura 6.24 é possível se visualizar o comportamento das aproximações das Frentes de Pareto quando se aumenta conjuntamente os requisitos e dependências. Nesse cenário, o menor índice de aproximação é de 93,70% (caso do *dataset-2* original) e o maior índice de aproximação é de 96,29%. Observa-se, então, uma tendência de aumento progressivo da qualidade do NSGA-II em relação às soluções ótimas a medida que o número de requisitos aumenta proporcionalmente com o número de dependências.

O próximo cenário mostra o comparativo de qualidade das soluções a medida que o número de requisitos aumenta. Os resultados podem ser vistos na Tabela 6.20 e no gráfico de tendências apresentado na Figura 6.25. É possível perceber que as taxas de aproximação do NSGA-II continuam elevadas. No entanto, estas taxas não são tão altas quanto no cenário anterior. De forma oposta ao cenário anterior, o *dataset-2* original possui a maior taxa de aproximação das Frentes de Pareto neste segundo cenário. A taxa de aproximação varia de 87,71% até 93,70% entre as instâncias analisadas, demonstrando uma tendência de redução leve a medida que o número de requisitos cresce.

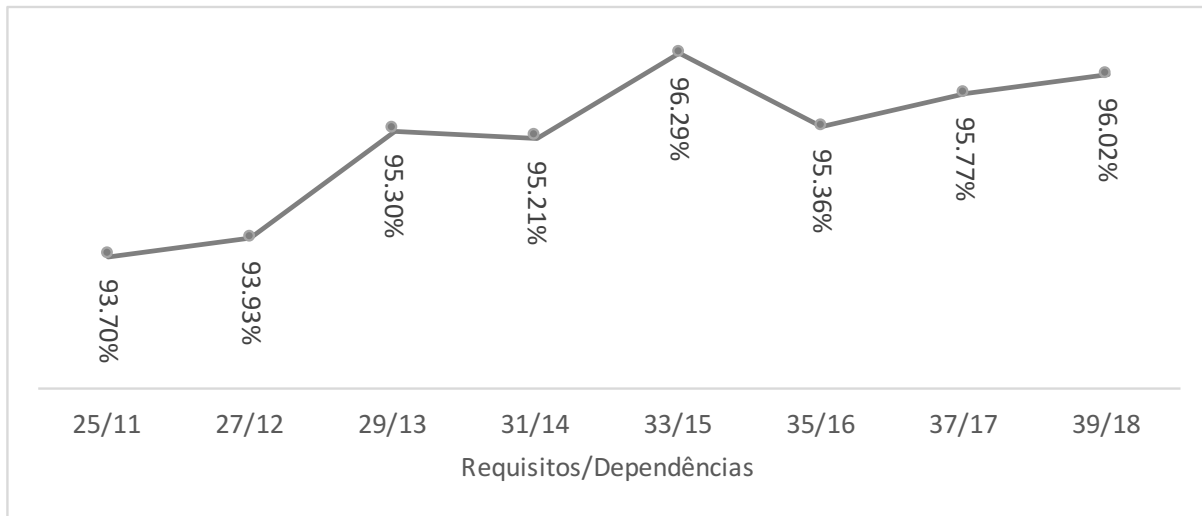


Figura 6.24: Tendência percentual para aproximação das Frentes de Pareto (dataset-2 escalabilidade)

Tabela 6.20: Resultados para o teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao aumento do número de requisitos (REQ)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg. Soluções Iguais	Avg aproximação %
dataset-2	143	170	134	93,70%
(27)	171	193	150	87,71%
(29)	222	252	198	89,18%
(31)	251	252	225	89,64%
(33)	271	304	240	88,56%
(35)	293	324	260	88,73%

Os últimos resultados dizem respeito ao cenário em que apenas o número de dependências aumenta gradativamente. A Tabela 6.21 mostra os resultados consolidados e a Figura 6.26 mostra o gráfico de tendências. Nesse cenário, é possível observar um crescimento da taxa de aproximação até as instâncias com 17 dependências, seguido por um decréscimo brusco na taxa de aproximação para instâncias com mais de 17 dependências. A perda de qualidade em relação ao Z3 pode ser claramente observada no gráfico de tendências a partir da instância com 19 dependências. Assim como nos resultados anteriores, vemos que a técnica heurística funciona melhor em espaços de busca mais livres, se afastando do SMT *solver*, em termos de qualidade das soluções encontradas, a medida que o espaço de busca se torna mais restrito.

6.4 DISCUSSÃO DOS RESULTADOS

RQ1. *É possível encontrar soluções que reduzam o risco da insatisfação dos stakeholders com impacto mínimo no custo e valor dos requisitos?*

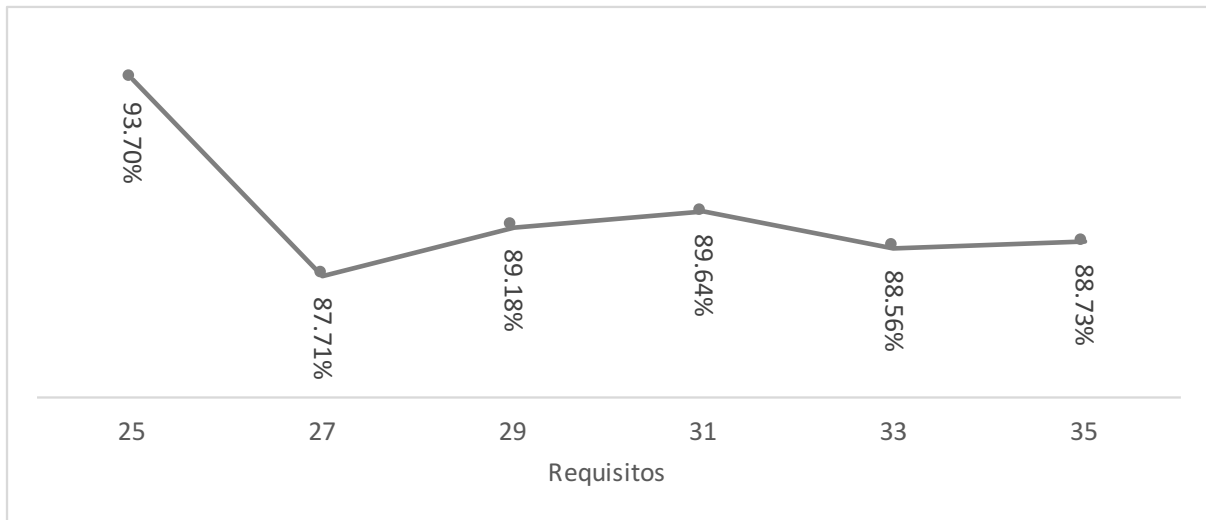


Figura 6.25: Tendência percentual para aproximação das Frentes de Pareto (dataset-2 escalabilidade)

Tabela 6.21: Resultados para teste de aproximação das Frentes de Pareto - Taxa percentual média de aproximação do NSGA-II em relação ao Z3 quanto ao aumento do número de dependências (DEP)

Dataset	Soluções-Z3	Soluções-NSGA-II	Avg. Soluções Iguais	Avg aproximação %
dataset-2	143	170	134	93,70%
(12)	137	160	130	94,89%
(13)	132	145	125	94,69%
(14)	126	152	114	90,47%
(15)	126	162	126	88,09%
(16)	127	144	120	94,48%
(17)	141	156	134	95,03%
(18)	118	143	108	91,52%
(19)	110	141	75	68,18%
(20)	110	142	75	68,18%
(21)	111	142	74	66,66%
(22)	109	143	78	71,55%
(23)	111	145	73	65,76%
(24)	99	133	63	63,63%
(25)	99	134	63	65,30%

As soluções produzidas pelo Z3 para o RA-MONRP, em ambos os conjuntos de dados e em todas as regiões de interesse, mostraram que uma redução substancial do risco de insatisfação dos *stakeholders* somente pode ser alcançada com impacto não negligenciável no custo e valor das soluções. Porém, em todos os cenários analisados, isto é, com margens de tolerância fixas e iguais (Tratamento 1) ou com margens de tolerância diferentes para custo e valor (Tratamentos 2 e 3), é sempre possível obter soluções que diminuem o risco com impacto limitado no custo e valor. Algumas soluções interessantes foram selecionadas

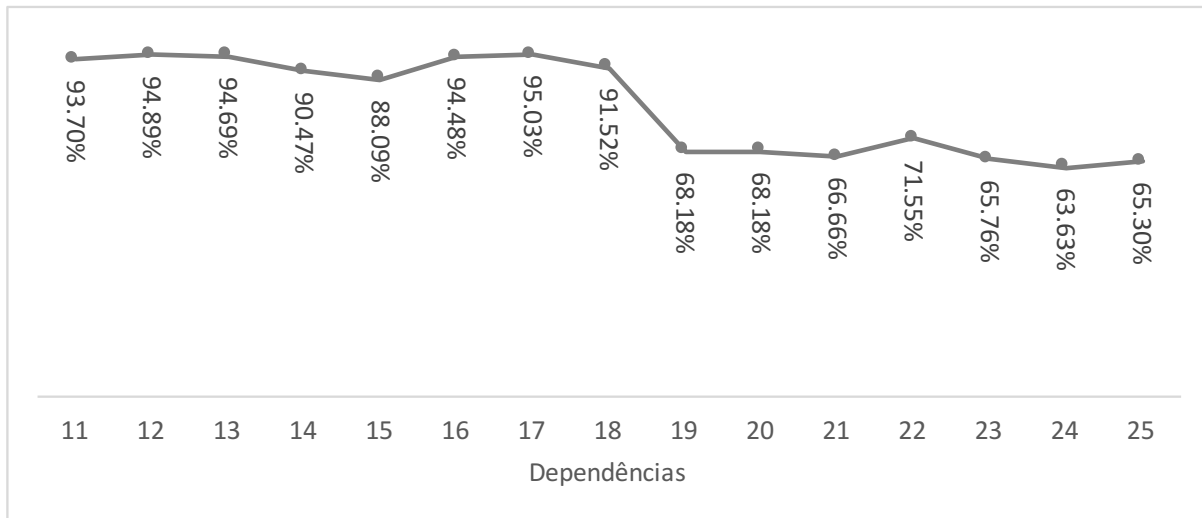


Figura 6.26: Tendência percentual para aproximação das Frentes de Pareto (dataset-2 escalabilidade)

e discutidas, apresentando tanto redução do risco quanto do custo, principalmente no cenário em que um dos objetivos não era limitado por margem de tolerância.

Em termos de número de soluções obtidas por região de interesse, o Tratamento 2 foi o que apresentou o maior número de soluções em cada região, seguido pelo Tratamento 3 e, por fim, pelo Tratamento 1. Esse resultado deve-se à característica aplicada em cada tratamento. Tanto o Tratamento 1 quanto o Tratamento 3 apresentavam regiões de interesse bem delimitadas para os objetivos, o que, por correspondência, limita o espaço de busca por soluções. Já no Tratamento 2, sempre havia um dos objetivos que podia variar livremente (ou seja, não estava limitado por uma margem de tolerância) e o espaço de busca era ampliado por esse aspecto.

De forma específica para o Tratamento 2, quando o objetivo relacionado ao valor dos requisitos não estava limitado por uma margem de tolerância, quase sempre as soluções com maiores reduções do risco vinham acompanhadas de perdas altas de valor. De forma oposta, as poucas soluções que apresentavam aumento de valor possuíam baixa redução do risco e/ou apresentavam aumento significativo do custo. De fato, risco e custo são objetivos que visam a minimização, enquanto o valor visa a maximização. Portanto, por serem objetivos conflitantes é plausível o comportamento de perdas/ganhos entre eles.

No que diz respeito ao tempo de execução para a busca de soluções RA-MONRP, mais uma vez, o Tratamento 2 apresentou os maiores tempos de busca por região de interesse, se comparado aos obtidos pelos Tratamentos 1 e 3. Exemplo disso, o Tratamento 2 apresentou duas situações em que não foi possível executar a busca dentro do tempo limite (*timeout*) estabelecido.

RQ2. *Qual a escalabilidade da abordagem quando o cenário cresce em número de requisitos, em números de dependências ou em ambos?*

Em relação às técnicas SMT utilizadas, o Z3 mostrou um desempenho superior em ter-

mos de tempo de execução quando comparado ao Yices. Porém, na maioria dos cenários de escalabilidade, o Z3 foi superado em tempo de execução pela meta-heurística NSGA-II.

Nos cenários em que o número de requisitos e dependências são aumentados conjuntamente foi possível perceber que o tempo de execução do Z3 aumenta rapidamente, enquanto o NSGA-II exibe apenas um leve decréscimo de desempenho em relação ao tempo de execução. Portanto, é possível afirmar que quando os dois atributos (requisitos e dependências) são aumentados proporcionalmente, o Z3 torna-se rapidamente inaplicável devido ao tempo excessivo que é necessário para concluir o processo de busca.

No que diz respeito ao crescimento apenas do número de requisitos, de forma similar à configuração anterior, o NSGA-II permaneceu mais rápido que o Z3. Nesse cenário, de instância para instância, o tempo de execução do Z3 cresce exponencialmente ao número de requisitos, enquanto o NSGA-II apresentou um crescimento leve entre as instâncias.

O último cenário de escalabilidade está relacionado com o aumento apenas do número de dependências, permanecendo constante o número de requisitos entre as instâncias. Para essa configuração, quase que a cada instância gerada, o Z3 necessitava de menos tempo para obter a Frente de Pareto, enquanto que o NSGA-II em algumas situações apresentava um pequeno aumento do tempo de execução de uma instância para outra. No caso do Z3, quando essa técnica alcançou uma relação entre requisitos e dependências de 0,76, o processo de busca foi realizado com um desempenho (tempo de execução) mais eficiente do que o NSGA-II.

Com base nos resultados obtidos no teste de escalabilidade, é possível afirmar que apenas o número de requisitos não é o suficiente para se determinar o limite da aplicabilidade de técnicas SMT, uma vez que a densidade das dependências também desempenha um papel importante para se determinar se um SMT terminará ou não o processo de busca em tempo aceitável. Em outras palavras, o limite de aplicabilidade de uma técnica SMT depende tanto do número de requisitos (um maior número de requisitos sendo associado com mais tempo de execução) quanto da densidade das dependências (maior densidade sendo associada a um menor tempo de execução).

RQ3. *Quão próximas são as soluções encontradas pela implementação da meta-heurística e a implementação baseada em SMT?*

Os primeiros resultados encontrados na avaliação da qualidade das soluções dizem respeito a comparação entre o Yices e o Z3 para os conjuntos de dados originais. Nesse caso, ficou demonstrado que o Yices incluía nos resultados algumas soluções denominadas *weak Pareto*, ou seja, existiam soluções em que os valores eram diferentes para um dos objetivos, porém iguais para o outro objetivo em questão (COELLO; LAMONT; VELDHUIZEN, 2007). De forma diferente, o Z3 apresentava a Frente de Pareto real. Logo, essa característica configura uma vantagem do Z3 em relação ao Yices, no que diz respeito à qualidade das soluções obtidas.

Na comparação entre o NSGA-II e o Z3 para as instâncias originais, as diferenças das taxas de aproximação são bastante evidentes. Para o *dataset-2*, o percentual médio de soluções do NSGA-II equivalentes à Frente de Pareto do Z3 é em torno de 93,70%. Por outro lado, o mesmo não ocorre em relação ao *dataset-1*. Nesse caso, a taxa de

aproximação entre as soluções do NSGA-II e do Z3 foi de apenas 42,80%, ou seja, menos de metade das soluções encontradas eram ótimas.

Para as instâncias referentes ao crescimento conjugado de requisitos e dependências, as soluções sub-ótimas incluídas pelo NSGA-II em suas Frentes de Pareto apresentavam diferença percentual pequena em relação à Frente de Pareto do Z3. Tanto em relação ao custo quanto ao valor, essas diferenças percentuais ficaram em média abaixo de 1% e a máxima diferença percentual também ficou abaixo deste valor, o que significa que, apesar de serem soluções sub-ótimas, são soluções adjacentes às soluções ótimas.

Por outro lado, todas as instâncias que representam aumento apenas no número de requisitos apresentaram boas taxas médias de aproximação de soluções entre as técnicas (acima de 87%), porém inferiores à taxa média de aproximação quando as instâncias são modificadas tanto no número de requisitos quanto no número de dependências. As diferenças médias entre as soluções permaneceram bastante baixas, caracterizando mais uma vez uma adjacência entre as Frentes de Pareto. Porém, foi possível perceber que as diferenças máximas também foram maiores que no cenário anterior, o que representa que algumas soluções se distanciavam da solução ótima mais próxima.

O cenário com aumento apenas do número de dependências foi o que apresentou a maior variação das taxas médias de aproximação entre as instâncias geradas: a maior taxa de aproximação obtida foi de 95,01% e a menor de 65,30%. Portanto, a eficácia do NSGA-II em obter soluções ótimas decaiu em espaços de busca mais restritos.

6.4.1 Análise Consolidada sobre as Questões de Pesquisa

Com os resultados obtidos é possível traçar algumas orientações para os *stakeholders* na tomada de decisão para a seleção de requisitos de software de acordo com seus desejos e expectativas.

Os tipos de tratamento apresentados para o RA-MONRP representam cenários interessantes para a obtenção de soluções que reduzam o risco. O Tratamento 1, por exemplo, representa o caso em que os *stakeholders* têm pouco conhecimento sobre os tipos de soluções (conjunto de requisitos) que encontrará para iniciar o processo decisório e necessitem de uma base de conhecimento inicial sobre as possíveis soluções e seus riscos. Nesse caso, com as margens de tolerância pequenas é possível identificar soluções RA-MONRP e decidir sobre o quanto expandir as margens para se obter mais resultados.

O Tratamento 2 é recomendado para empresas de software em que os *stakeholders* desejam ter uma maior número de alternativas para a seleção dos requisitos para a próxima versão do software. Nesse caso, o tempo de execução da busca será muito maior e o processo decisório poderá requerer mais tempo, dado que haverá um maior número de soluções para serem analisadas. No entanto, é uma estratégia interessante para formar uma visão mais próxima do conjunto completo de soluções dado que, quando um objetivo não possui restrição imposta, este fato amplia o número de soluções obtidas.

O Tratamento 3 é recomendado para situações em que os *stakeholders* têm conhecimento estabelecido em relação aos requisitos envolvidos no projeto e quanto aos objetivos de custo e valor relacionado à próxima versão. De posse de margens conhecidas para estes objetivos, é possível estabelecer as regiões de interesse de forma mais precisa, visando

obter soluções dentro dos objetivos da empresa desde as primeiras análises.

Um aspecto importante é que, quanto maior o número de *stakeholders*, mais importante será a aplicação do RA-MONRP para dirimir dúvidas sobre as diferenças entre as soluções a ser discutidas, pois mais *stakeholders* implicam em mais desejos, necessidades e níveis de importância atribuídos aos requisitos e, por consequência, maior variabilidade no conjunto de soluções a ser analisado. Se o processo de análise fosse realizado sem as soluções RA-MONRP em evidência, possivelmente a tomada de decisão seria mais longa por conta da insatisfação de alguns *stakeholders*.

Neste ponto está a maior utilidade do RA-MONRP, que é o destaque dado a soluções que reduzam a insatisfação dos *stakeholders*. Os resultados experimentais discutidos nas seções anteriores mostraram que, para uma solução de referência escolhida dentro de parâmetros definidos de custo e valor, existem soluções na vizinhança que apresentam redução de risco em relação a ela. Este resultado pode ter um impacto positivo no projeto, uma vez que a seleção de uma solução que não leve em consideração as demandas do conjunto de *stakeholders* pode implicar em retenção de conhecimento e falta de apoio dos *stakeholders* insatisfeitos (LAMSWEERDE, 2009; FEATHER; CORNFORD, 2003; ASNAR; GIORGINI; MYLOPOULOS, 2011; MOORES; CHAMPION, 1996; MCNEIL; FREY; EMBRECHTS, 2005; HAN; HUANG, 2007).

Diante dos dados obtidos foi possível organizar os resultados entre o tempo de execução e qualidade das soluções (Tabelas 6.22, 6.23, 6.24 e 6.25), visando dar suporte à escolha da técnica de busca para a seleção dos requisitos de software para a próxima versão. Para os conjuntos de dados originais (Tabela 6.22, os resultados põem em evidência duas situações distintas. Para o *dataset-1*, que possui um espaço de busca maior devido ao número de requisitos, o tempo de execução do Z3 é muito maior do que o tempo médio exigido pelas execuções do NSGA-II. No entanto, na média, a aproximação dos conjuntos de soluções obtidos pelo NSGA-II em relação ao Z3 está abaixo de 50%. Por outro lado, para o *dataset-2*, a taxa de aproximação do NSGA-II em relação à Frente de Pareto do Z3 é alta, atingindo cerca de 93,70% de intersecção entre os conjuntos de soluções obtidos pelos dois processos de busca. Nessa última situação, o NSGA-II também é mais rápido, porém não tão acentuadamente quanto para o *dataset-1*.

Tabela 6.22: Dados agrupados para tempo de execução e taxa de aproximação para os conjuntos de dados originais

Dataset	Tempo (Z3)	Tempo médio (NSGA-II)	Taxa de aproximação (%)
dataset-1	183.320,89	50,25	42,80%
dataset-2	81,86	21,95	93,70%

As Tabelas 6.23, 6.24 e 6.25, mostram os dados consolidados quando o teste de escalabilidade foi realizado, ressaltando a relação entre os tempos de execução e a taxa de aproximação. Nessas tabelas é possível identificar que o NSGA-II é mais rápido que o Z3 na maioria dos casos e que, em situações nas quais a densidade de dependências é menor do que 0,72, as Frontes de Pareto do NSGA-II se aproximam muito dos resultados do Z3. Quando a densidade é superior a 0,76, o Z3 é melhor em tempo de execução e o NSGA-II

Tabela 6.23: Dados agrupados para tempo de execução e taxa de aproximação para o aumento conjunto de requisitos e dependências

Reqs/Deps	Tempo (Z3)	Tempo médio (NSGA-II)	Taxa de aproximação (%)
(27/12)	273,19	24,26	93,93%
(29/13)	960,83	28,59	95,30%
(31/14)	3.410,90	29,89	95,21%
(33/15)	8.342,45	32,50	96,29%
(35/16)	36.175,45	37,85	95,36%
(37/17)	77.336,76	46,61	95,77%
(39/18)	203.239,2	49,68	96,02%

Tabela 6.24: Dados agrupados para tempo de execução e taxa de aproximação para o aumento apenas do número de requisitos

Reqs/Deps	Tempo (Z3)	Tempo médio (NSGA-II)	Taxa de aproximação (%)
(27/11)	418,38	23,60	87,71%
(29/11)	1.952,61	32,38	89,18%
(31/11)	8.047,95	34,39	89,64%
(33/11)	30.136,87	42,00	88,56%
(35/11)	168.452,52	46,72	88,73%

Tabela 6.25: Dados agrupados para tempo de execução e taxa de aproximação quando do aumento apenas do número de dependências

Reqs/Deps	Tempo (Z3)	Tempo médio (NSGA-II)	Taxa de aproximação (%)
(25/12)	48,58	22,49	94,89%
(25/13)	39,58	19,73	94,69%
(25/14)	33,81	20,30	90,47%
(25/15)	27,67	20,91	88,09%
(25/16)	26,49	18,86	94,48%
(25/17)	26,73	21,06	95,03%
(25/18)	22,67	18,67	91,52%
(25/19)	15,73	18,50	68,18%
(25/20)	10,66	18,57	68,18%
(25/21)	7,88	18,72	66,66%
(25/22)	3,34	18,86	71,55%
(25/23)	2,96	18,77	65,76%
(25/24)	2,15	18,09	63,63%
(25/25)	1,96	18,06	65,30%

vai diminuindo gradativamente sua eficiência em relação à Frente de Pareto ótima.

A partir das três questões de pesquisa e dos resultados obtidos, é possível dizer que os *stakeholders* podem se deparar com duas situações para a tomada de decisão de soluções que reduzam o risco de insatisfação dos *stakeholders*: urgência e precisão das soluções.

O Z3, por ser uma técnica determinística, apresenta melhor desempenho nesse último atributo, mas por outro lado quase sempre o processo de busca é mais demorado. Por outro lado, o NSGA-II quase sempre é mais rápido, porém pode acrescentar soluções sub-ótimas no processo de busca. Dessa forma, recomenda-se que os *stakeholders* efetuem inicialmente o cálculo da densidade da instância para depois escolher qual técnica será selecionada (determinística ou meta-heurística). Limitações de hardware e processador podem interferir nessa decisão, já que quanto mais potente for o processador utilizado, mais rápido tende a ser o processo de busca.

Feito isto e escolhidas as soluções que servirão de ponto de partida, recomenda-se que uma técnica determinística seja aplicada no cálculo ao RA-MONRP para que se evite a inclusão de possíveis soluções sub-ótimas (como no caso de uma metaheurística). Em termos gerais, sempre que possível e de acordo com os cenários expostos nesse trabalho, recomenda-se que os usuários dessa abordagem apliquem um *SMT Solver* e recorram a uma metaheurística apenas se o *SMT Solver* não execute ou mostre a tendência a não terminar a execução num tempo previamente estabelecido como aceitável pelos *stakeholders*. Essa relação é dependente do contexto da companhia de software pois uma empresa pode estabelecer como limite aceitável menos de um minuto para obter soluções RA-MONRP em uma Região de Interesse, enquanto outra pode aceitar até algumas horas para se obter a Frente de Pareto determinística.

6.5 AMEAÇAS À VALIDADE

Os resultados deste trabalho estão sujeitos às limitações típicas de qualquer estudo empírico. Esta seção define as ameaças à validade dos resultados experimentais, indicando como elas podem afetar a generalização dos resultados obtidos. Seguindo o delineamento proposto por Wohlin et al (WOHLIN et al., 2012) e Wieringa (WIERINGA, 2014), quatro categorias de ameaças podem existir no processo experimental. São elas a ameaça de validade externa, de construção, interna e de conclusão.

No que diz respeito à validade externa, ou seja, em relação à generalização dos resultados dos experimentos para outros contextos, dois conjuntos de dados reais foram usados nos experimentos conduzidos na abordagem proposta para o RA-MONRP, o que mostra um indicativo de que tais resultados poderiam ser aplicados em ambientes similares da indústria de software. No entanto, não é possível afirmar que os resultados obtidos podem ser generalizados para ambientes arbitrariamente complexos no que diz respeito aos requisitos de software. Nesse caso, mais experimentos com configurações de dados similares e com instâncias muito maiores devem ser conduzidos para se verificar a aplicabilidade das técnicas usadas na abordagem experimental.

A validade de construção está relacionada às métricas usadas nos experimentos. Os atributos estudados foram requisitos e as propriedades associadas a cada requisito (custo, valor e risco). Custo e valor são medidas vindas das estimativas realizadas pelos *stakeholders* e disponibilizadas nos conjuntos de dados originais. O risco foi calculado para capturar a incerteza nas estimativas feitas por múltiplos *stakeholders*. Consequentemente, os resultados baseiam-se em métricas de qualidade associadas ao cálculo do risco. Nesse caso, a variação percentual foi usada como métrica para avaliar o *trade-off* alcançado

pela abordagem usada nos experimentos.

Para avaliar a escalabilidade, foram comparados os tempos de execução de cada técnica. Para avaliar a eficácia e eficiência da meta-heurística, os conjuntos de soluções obtidos por essa técnica foram comparados aos conjuntos de soluções obtidos pela técnica determinística, medindo-se a taxa de aproximação para as Frentes de Pareto de uma mesma instância e as diferenças percentuais de custo e valor. No que diz respeito à instrumentação, foram utilizadas ferramentas de código aberto e implementações de algoritmos amplamente testadas pela comunidade acadêmica. Essas ferramentas e os procedimentos para o processamento de dados foram extensivamente testados antes de aplicá-los nos conjuntos de dados usados nos experimentos.

Relativo à validade interna, o aspecto principal dessa ameaça está relacionado com os parâmetros das técnicas utilizadas, principalmente a meta-heurística. Para atenuar essa ameaça, os operadores do NSGA-II, assim como os valores de parâmetros, foram sempre os mesmos em todas as execuções das diferentes instâncias. Da mesma forma, a técnica determinística sempre foi aplicada com a mesma configuração, obedecendo aos critérios relativos à base de conhecimentos da *Satisfiability Modulo Theory*. Para as técnicas de busca utilizadas, cada código-fonte foi disponibilizado e teve os detalhes de implementação explicados, identificando as entradas e saídas de cada algoritmo.

Em relação à ameaça de conclusão, a ausência de trabalhos semelhantes ao RA-MONRP que utilizem SMT *solvers* implica em uma dificuldade de comparar os resultados encontrados com outras abordagens que utilizem esse tipo de técnica de busca determinística. No entanto, acredita-se que as técnicas utilizadas e os resultados obtidos indicam uma perspectiva factível para aplicação do RA-MONRP em contextos realistas.

Em relação à variação aleatória nos algoritmos é importante ressaltar que o NSGA-II é um algoritmo não-determinístico. Logo, o mesmo foi executado 100 vezes para cada configuração e foi relatado seu desempenho médio. A partir das execuções para cada instância do NSGA-II, as médias de tempo de execução e taxas de aproximação foram sumarizadas em tabelas para permitir a comparação em todas as instâncias. O Z3 foi executado apenas uma vez para cada instância devido a ser um algoritmo determinístico, o que faz com que sempre apresente os mesmos resultados quando é executado para uma determinada instância.

6.6 CONSIDERAÇÕES FINAIS

Este capítulo mostrou a avaliação da abordagem proposta no capítulo anterior, detalhando a metodologia e o procedimento adotados para a avaliação, a análise dos resultados e as ameaças à validade dos estudos. Os resultados foram apresentados em tabelas, histogramas e gráficos de tendência para auxiliar a visualização e interpretação. Ao final, foi realizada uma discussão dos resultados obtidos de acordo com cada questão de pesquisa para dar embasamento à tomada de decisão na redução do risco de insatisfação dos *stakeholders*.

O próximo capítulo apresenta a conclusão desse trabalho de Tese, bem como suas limitações e potenciais trabalhos futuros que podem ser desenvolvidos a partir do ponto em que encerramos a proposta e sua avaliação.

CONCLUSÃO E TRABALHOS FUTUROS

Uma atividade crítica no desenvolvimento de software é o planejamento da próxima versão do software. Tal atividade visa selecionar um subconjunto dos requisitos a ser implementado na próxima versão, maximizando o valor entregue para os *stakeholders* e minimizando o custo para desenvolver a nova versão do software. Há, no entanto, uma terceira dimensão que deve ser considerada: o risco inerente a diferentes estimativas de valor feitas pelos múltiplos *stakeholders*.

Quando a variabilidade das estimativas de valor é elevada, aumenta a probabilidade de que o valor efetivamente entregue para os *stakeholders* afaste-se do valor médio, que é comumente usado para o planejamento da próxima versão. Neste trabalho, esse risco foi denominado de *risco da insatisfação dos stakeholders* e uma abordagem foi apresentada para minimizar tal risco, visando um impacto limitado no custo e valor.

A abordagem proposta integra o risco de insatisfação dos *stakeholders* a uma formulação multi-objetivo (RA-MONRP), aplicando uma técnica de busca determinística em regiões de interesse para encontrar soluções que minimizem o custo, maximizem o valor da solução e reduzam o risco de insatisfação dos *stakeholders*. Os resultados obtidos mostraram que:

- O risco de insatisfação dos *stakeholders* pode ser minimizado com impacto não negligenciável no custo e no valor. Além disso, soluções interessantes foram encontradas em que a minimização do risco não afeta negativamente o custo, ou seja, existem casos em que o risco decrescia e o custo era reduzido, embora esse tipo de solução exija uma redução maior no valor entregue;
- A abordagem proposta é escalável quando o NSGA-II é usado. O SMT *solver* (Z3) apresenta escalabilidade em problemas que não são excessivamente grandes em termos de número de requisitos e/ou não são escassos em termos de dependências. Observamos que a escalabilidade é negativamente afetada pelo tamanho do problema (número de requisitos), mas também por uma baixa densidade de dependências. No geral, o SMT *solver* parece encontrar soluções mais rapidamente em espaço de busca restrito por um grande número de dependências;

- Na maioria das situações, as soluções encontradas pelo NSGA-II são apenas marginalmente abaixo do ideal. Assim, essa meta-heurística pode ser utilizada como um substituto para os SMT *solvers* quando estes não são capazes de encontrar soluções em um tempo razoável. Apenas nos casos em que um conjunto de dados apresenta alta densidade de dependências é que o NSGA-II se distancia consideravelmente da Frente de Pareto ótima.

Com base nos resultados obtidos, recomenda-se que o risco de insatisfação dos *stakeholders* seja integrado às formulações multi-objetivo para seleção de requisitos de software e que, sempre que possível, uma técnica de busca determinística (por exemplo, um SMT *solver*) seja utilizada na otimização para obter a Frente de Pareto. Se uma técnica determinística não apresentar os resultados em um tempo computacional satisfatório para os usuários do RA-MONRP, uma meta-heurística pode ser aplicada para obter soluções próximas à Frente de Pareto com um tempo de execução menor.

7.1 LIMITAÇÕES DO TRABALHO

Os objetivos definidos na Tese foram alcançados de acordo com a abordagem e validação propostas. Neste sentido, foi apresentada a formulação multi-objetivo RA-MONRP que considera o risco de insatisfação dos *stakeholders*, duas abordagens de busca exata baseadas em SMT *solvers* foram implementadas para encontrar soluções para esta formulação, uma abordagem baseada no algoritmo NSGA-II foi implementada com o mesmo objetivo e foram realizados estudos experimentais para comparar as três técnicas.

Entretanto, algumas limitações podem ser verificadas nesse trabalho. A primeira delas diz respeito à aplicação da abordagem em ambientes de uma ou mais companhias de software, o que aumentaria a confiança na validade da proposta. Apesar dos conjuntos de dados utilizados serem advindos de projetos reais, a aplicação da abordagem é passível de evidências mais fortes quando há a presença de fato dos *stakeholders* ao longo de todo o processo de otimização.

Outra limitação diz respeito à obtenção de conjuntos de dados reais. Isso se configura como uma tarefa difícil, seja porque as empresas de software não desejam disponibilizar seus dados ou os classificam como confidencial, ou porque esses dados são incompletos. Por exemplo, algumas empresas de software que foram contatadas se disponibilizaram a ceder seus dados de projetos, porém não apresentavam, por exemplo, a importância dos *stakeholders* e/ou as interdependências entre os requisitos. Caso mais bases de dados reais fossem obtidas, a abordagem poderia ser aplicada em mais cenários, ampliando o escopo de análise e validação dos resultados.

Outra limitação diz respeito à utilização de mais técnicas de busca no processo de experimentação. O ideal seria o uso de mais SMT *solvers* e meta-heurísticas, o que possibilitaria um cenário com mais comparações entre as técnicas, visando avaliar o tempo de execução de cada uma, as frentes de Pareto obtidas e o comportamento para o teste de escalabilidade. No entanto, tanto o NSGA-II quanto o Z3 são técnicas bastante testadas e aplicadas em problemas multi-objetivo. A primeira delas é base de referência para processos de busca e pesquisas realizadas pela comunidade científica de SBSE e a segunda

para a comunidade de pesquisa operacional. A seleção de ambas para o processo de experimentação proposto foi realizada através de pesquisa nessas áreas.

7.2 TRABALHOS FUTUROS

Em relação aos trabalhos futuros que podem ser realizados a partir dos resultados apresentados neste trabalho, tanto a própria abordagem proposta quanto a formulação podem ser expandidos. Por exemplo, se a variabilidade associada com as estimativas de custo também é um fator de risco, o RA-MONRP pode ser estendido de três para quatro objetivos, com o intuito de medir o desvio padrão ponderado das estimativas de custo dos desenvolvedores, ou seja, inclusão de um novo objetivo que leve em consideração o fator de risco associado ao custo de desenvolvimento dos requisitos de software.

Além disso, restrições adicionais podem ser aplicadas ao RA-MONRP, como orçamento máximo de custo, o valor mínimo a ser entregue, o risco máximo aceitável ou restrições relacionadas com a seleção de regiões de interesse, especificando um limite inferior e um limite superior para cada objetivo.

Sob o aspecto de visualização dos resultados, a análise de *clusters* é uma possibilidade que será experimentada para prover melhor suporte na interpretação das soluções, agrupadas em subconjuntos de similaridade. Estudos iniciais estão sendo realizados para a configuração e incorporação de diferentes tipos de *cluster* vindos da área de bioinformática, aplicando diferentes visualizações que podem ser úteis ao contexto específico da empresa de software, tais como, *radial cladogram*, *radial phylogram*, *circular phylogram* e *rectangular cladogram* (LINZ; JOHN; SEMPLE, 2013; LIMA, 2014).

Em adição aos dois trabalhos futuros citados acima, uma possibilidade importante para a validação da abordagem é a realização de estudos empíricos incluindo o ser humano (*stakeholders*) no processo de otimização para avaliar a usabilidade e aceitabilidade da abordagem utilizada na Tese. Para tanto, se planeja avaliar o papel dos *stakeholders* interativamente (*human-in-the-loop*) na seleção das regiões de interesse, margens de tolerância e seleção do subconjunto de requisitos de software. Em outros termos, essa proposta de trabalho futuro visa inserir as preferências do grupo de *stakeholders* ao longo de todo o processo de otimização e não apenas entre o MONRP e o RA-MONRP.

Como consequência do processo iterativo, uma nova formalização multi-objetivo será desenvolvida visando incluir os fatores de risco associados ao custo, risco da insatisfação do usuário e a minimização das discordâncias dos *stakeholders* em relação às dependências e escala de prioridades entre os requisitos de software. Para tanto, técnicas vindas da área de pesquisa operacional podem contribuir para tratar os conflitos de decisão quando há a presença de múltiplos grupos de interesse no projeto de software.

Por fim, uma proposta voltada para aprimorar o processo de busca seria a hibridização de duas técnicas SMT na otimização para melhorar o tempo de execução na busca dos conjuntos de soluções. Espera-se com isso que o uso conjunto de SMT *solvers* permita que se apliquem tais técnicas quando o número de requisitos seja maior do que os conjuntos de dados usados nesse trabalho.

ESTUDOS SELECIONADOS PARA A REVISÃO

Abaixo, estão relacionados os estudos selecionados e compilados como parte da revisão sistemática (PITANGUEIRA; MACIEL; BARROS, 2015).

A.1 ESTUDOS PUBLICADOS EM 2001

- 1) The Next Release Problem (BAGNALL; RAYWARD-SMITH; WHITTLE, 2001)

A.2 ESTUDOS PUBLICADOS EM 2003

- 2) Quantitative Studies in Software Release Planning under Risk and Resource Constraints (RUHE; TN; GREER, 2003)

A.3 ESTUDOS PUBLICADOS EM 2004

- 3) Software Release Planning: an evolutionary and iterative approach (GREER; RUHE, 2004)

A.4 ESTUDOS PUBLICADOS EM 2005

- 4) Supporting Software Release Planning Decision for Evolving Systems (SALIU; RUHE, 2005b)
- 5) Determination of the Next Release of a Software product: an Approach using Integer Linear Programming (AKKER et al., 2005a)
- 6) Software Release Planning for Evolving Systems (SALIU; RUHE, 2005a)
- 7) Flexible Release Planning using Integer Linear Programming (AKKER et al., 2005b)

A.5 ESTUDOS PUBLICADOS EM 2006

- 8) Search Based Approaches to Component Selection and Prioritization for the Next Release Problem (BAKER; HARMAN; STEINH, 2006)

A.6 ESTUDOS PUBLICADOS EM 2007

- 9) The Multi-Objective Next Release Problem(ZHANG; HARMAN; MANSOURI, 2007)
- 10) Bi-Objective Release Planning for Evolving Software (SALIU; RUHE, 2007)
- 11) Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product (LI et al., 2007)

A.7 ESTUDOS PUBLICADOS EM 2008

- 12) Software Product Release Planning Through optimization and what-if Analysis (AKKER et al., 2008)
- 13) A systematic approach for solving the wicked problem of software release planning (NGO-THE; RUHE, 2008)

A.8 ESTUDOS PUBLICADOS EM 2009

- 14) Search Based Data Sensitivity Analysis applied to Requirement Engineering (HARMAN et al., 2009)
- 15) A Study of The Multi-Objective Next Release Problem (DURILLO et al., 2009)
- 16) A New approach to The Software Release Planning (COLARES et al., 2009)
- 17) A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making (FINKELSTEIN et al., 2009)

A.9 ESTUDOS PUBLICADOS EM 2010

- 18) Search Based Optimization of Requirements Interaction Management (ZHANG; HARMAN, 2010)
- 19) Using Interactive GA for Requirements Prioritization (TONELLA; SUSI; PALMA, 2010)
- 20) An Integrated Approach for Requirement Selection and Scheduling in Software Release Planning (Li et al., 2010)
- 21) Ant Colony Optimization for the Next Release Problem (SAGRADO; Del Aguila; ORELLANA, 2010)
- 22) A Hybrid ACO Algorithm for The Next Release Problem (JIANG et al., 2010)
- 23) Approximate Backbone Based Multilevel Algorithm for Next Release Problem (JIANG; XUAN; REN, 2010)

A.10 ESTUDOS PUBLICADOS EM 2011

- 24) Comparing The Performance of Metaheuristics for the Analysis of Multi-stakeholder Tradeoffs in Requirements Optimisation (ZHANG et al., 2011)
- 25) Software Next Release Planning Approach Through Exact Optimization (G. Freitas; P. Coutinho; T. Souza, 2011)
- 26) A Study of The Bi-Objective Next Release Problem (DURILLO et al., 2011)
- 27) An Ant Colony Optimization Approach to the software Release Planning Problem with Dependent Requirements (SOUZA et al., 2011)

28) A Fuzzy Approach to Requirements Prioritization (LIMA et al., 2011)

A.11 ESTUDOS PUBLICADOS EM 2012

29) Software Requirements Selection using Quantum-inspired Elitist Multi-objective Evolutionary Algorithm (KUMARI; SRINIVAS; GUPTA, 2012)

30) Solving the Large Scale Next Release Problem with a Backbone-based Multilevel Algorithm (XUAN et al., 2012)

31) Evolutionary Approaches for Multi-Objective Next Release Problem (CAI; WEI; HUANG, 2012)

32) A Multiobjective Optimization Approach to the Software Release Planning with Undefined Number of Releases and Interdependent Requirements (BRASIL et al., 2012)

33) Multi-objective Optimization Approaches to software Release Time Determination (LI; XIE; NG, 2012)

A.12 ESTUDOS PUBLICADOS EM 2013

34) Empirical Evaluation of Search Based Requirements Interaction Management (ZHANG; HARMAN; LIM, 2013)

35) Interactive Requirements Prioritization using a Genetic Algorithm (TONELLA; SUSI; PALMA, 2013)

36) A Scenario-based Robust Model for the Next Release Problem (PAIXAO; SOUZA, 2013a)

37) A Recoverable Robust Approach for the Next Release Problem (PAIXAO; SOUZA, 2013b)

38) Hill Climbing and Simulated Annealing in Large Scale Next Release Problem (MAUŠA et al., 2013)

39) A Hybrid of Decomposition and Domination Based Evolutionary Algorithm for Multi-Objective Software Next Release Problem (CAI; WEI, 2013)

RESULTADOS ADICIONAIS PARA O TRATAMENTO 2

Nesse apêndice são apresentados os histogramas e gráficos de tendência das demais regiões de interesse para cada *dataset*, de acordo com os cenários das Tabelas 6.7 e 6.8.

O histograma e o gráfico de tendências para a região de interesse 2 do *dataset-1* podem ser vistos nas Figuras B.1 e B.2. Da mesma forma, como o número de soluções encontradas no processo de busca foi elevado (439 soluções), a análise das linhas de tendência e do histograma é melhor compreendida quando focada em uma parte menor do conjunto de soluções. Nesse cenário, é possível identificar que as soluções que mostram uma redução maior do risco, também são relacionadas com um aumento de custo consideravelmente alta. Tome-se como exemplo a solução 32, que apresenta a maior redução de custo (9,43%). Nesse caso, há um aumento acentuado do custo próximo a 12% em relação à solução de referência (solução 13), apesar de uma perda pequena de valor (1,67%). Seguindo esse mesmo comportamento, podem ser citadas as soluções 26, 29 e 33, todas apresentando uma redução do risco maior do 7,6%, porém com aumento do custo maior que 9,0%.

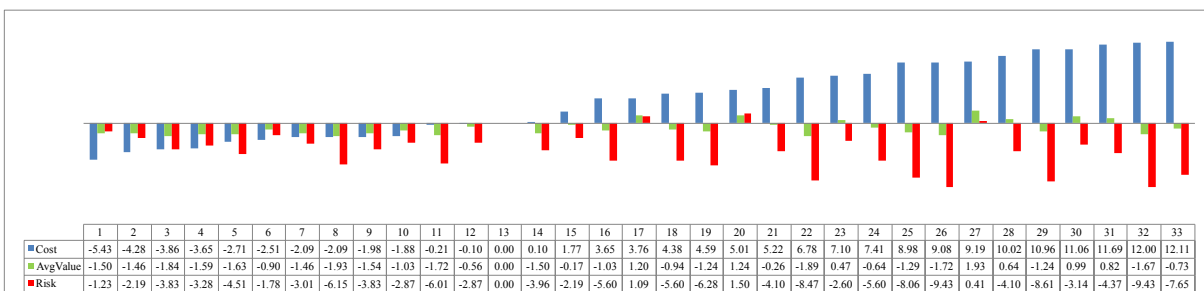


Figura B.1: Histograma de variações percentuais para a região de interesse 2 (*dataset-1*), com custo livre

Os próximos resultados dizem respeito à região de interesse 3 para o *dataset-1*. Para esse caso, a solução 27 é considerada como solução de referência e as demais soluções

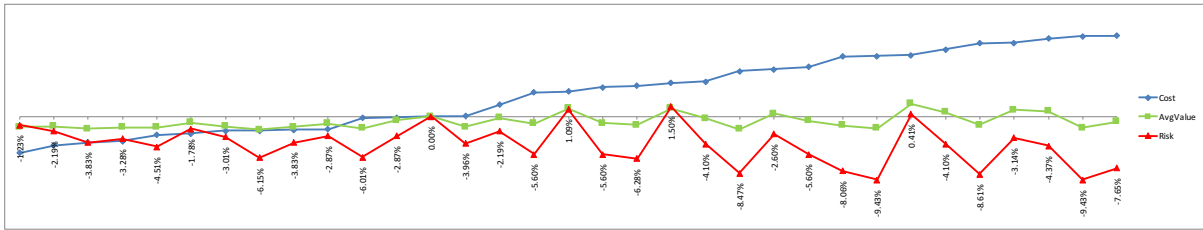


Figura B.2: Tendência de variações para a região de interesse 2 (*dataset-1*), com custo livre

fazem parte da vizinhança. É possível observar diversas soluções com redução de risco acima de 11%. O histograma da Figura B.3 põe em evidência uma amostra de soluções para as 206 que foram encontradas. Por exemplo, a solução 9 apresenta uma redução de risco de 12,82%, acompanhada de uma redução de custo de quase 3% e uma perda de valor de 6,24%. Existem também soluções que apresentam redução acentuada do risco e, ao mesmo tempo, uma redução pequena do custo (de fato, os dois objetivos não são sempre conflitantes), porém com uma perda de valor não negligenciável (por exemplo, próximo de 4% do valor). Esse é o caso das soluções 11 e 13. A solução 11 apresenta uma redução de risco próxima de 8,86%, uma redução de custo de 2,92% e uma perda de valor de 4,22%. De forma similar, a solução 13 mostra uma redução do risco em cerca de 10,26%, redução de custo em 2,77% e perda de valor próxima a 4,18%. O gráfico de tendência da Figura B.4 mostra a relação de perda/ganho para cada solução RA-MONRP.

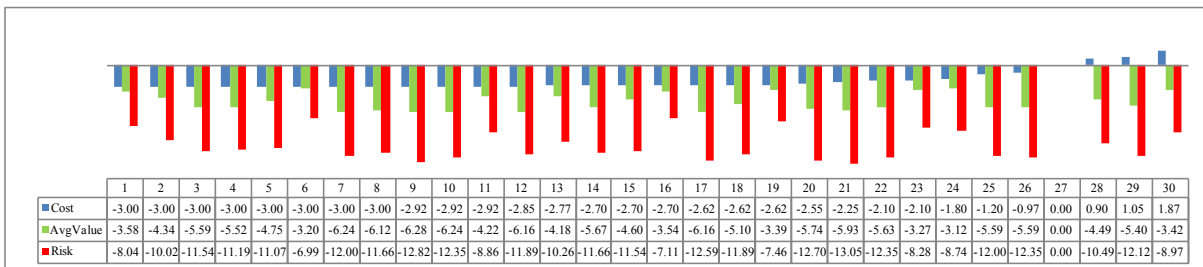


Figura B.3: Histograma de variações percentuais para a região de interesse 3 (*dataset-1*), com valor livre

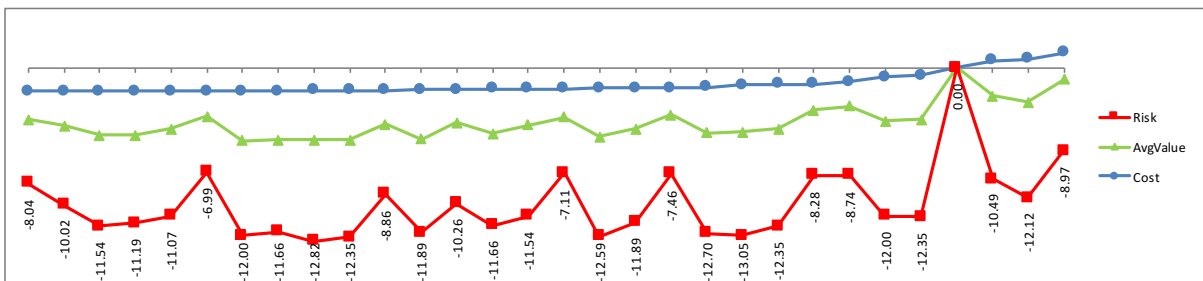


Figura B.4: Tendência de variações para a região de interesse 3 (*dataset-1*), com valor livre

Por fim, as Figura B.5 e B.6 mostra o histograma para 30 das 67 soluções encontradas para a região de interesse 3 com a configuração do custo livre.

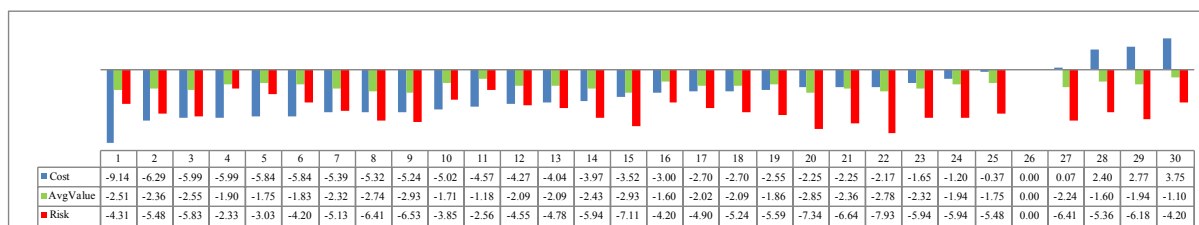


Figura B.5: Histograma de variações percentuais para a região de interesse 3 (*dataset-1*), com custo livre

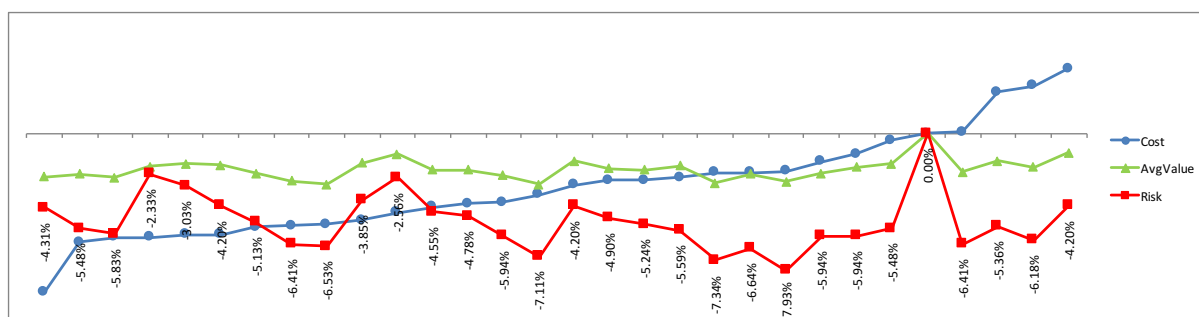


Figura B.6: Tendência de variações para a região de interesse 3 (*dataset-1*), com custo livre

Uma vez mais, algumas soluções interessantes foram encontradas, tais como solução 1, onde é possível reduzir o risco em 4,31% com uma perda de valor relativamente pequena (2,51%) e com uma redução de custo bastante acentuada (9,14%). A Figura põe em evidência o gráfico de tendências para a relação de perda/ganho das soluções RA-MONRP para a região de interesse 3 do *dataset-1* quando o custo pode variar livremente.

Em relação aos resultados obtidos para o *dataset-2* (ver Tabela 6.8), os resultados adicionais para a Região de Interesse 1 são mostrados a seguir. Inicialmente, a Figura B.7 mostra o histograma para essa configuração apresentando uma amostra de 30 soluções para as 235 obtidas e a Figura B.8 mostra o gráfico de tendências.

É possível identificar que diversas soluções apresentam redução de risco acima de 12%. No entanto, praticamente todas elas, apresentam de forma conjugada uma perda de valor alta. Para exemplificar, as soluções 18, 2 e 10 representam essa situação. No caso da primeira (solução 18), uma redução de risco em 16,56% é atingida relação à solução inicial (solução 20), com uma pequena redução de custo em torno de 0,22% e com uma perda significativa no valor próximo a 10,75%.

As próximas Figuras (B.9 e B.10) mostram os resultados para o *dataset-2* (ROI 2) quando o valor está livre de restrições e o custo está limitado pela margem de tolerância. Uma amostra de 35 soluções das 482 obtidas é mostrada. A solução 16 representa a solução inicial não possuindo, portanto, nenhum valor de variação de percentagem.

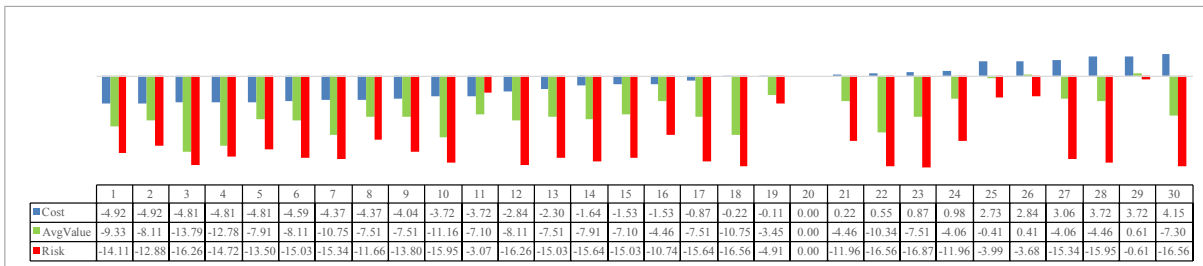


Figura B.7: Histograma de variações de percentagens para a região de interesse 1 (*dataset-2*), com valor livre

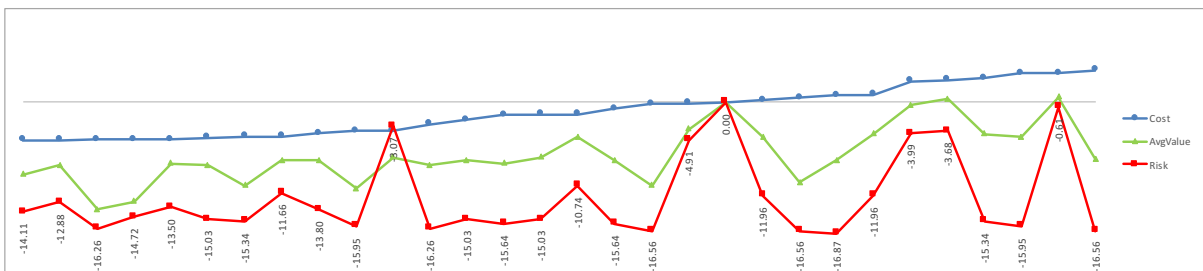


Figura B.8: Tendência de variações para a região de interesse 1 (*dataset-2*), com valor livre

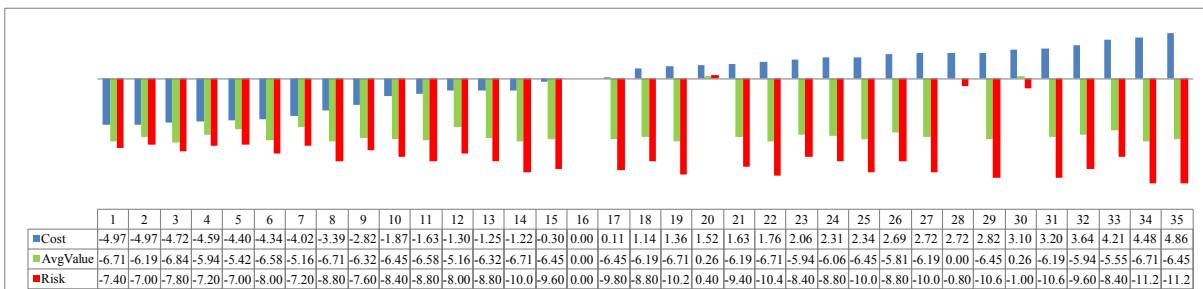


Figura B.9: Histograma de variações de percentagens para a região de interesse 2 (*dataset-2*), com valor livre

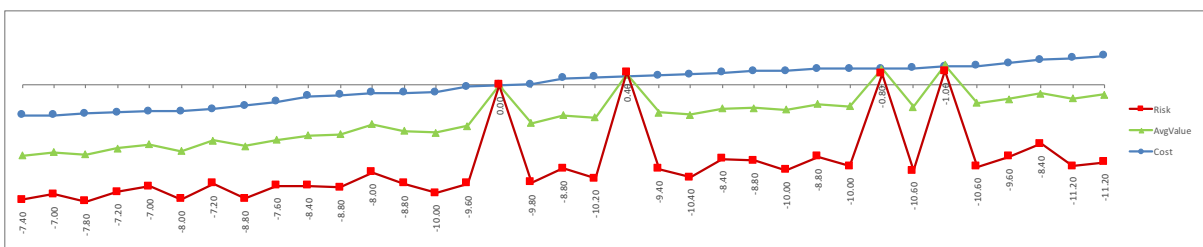


Figura B.10: Tendência de variações para a região de interesse 2 (*dataset-2*), com valor livre

Nesse cenário é possível identificar a solução 35, que apresenta um valor de redução de risco próximo a 11,2%. Porém, para tanto, a obtenção dessa redução do risco mostra

um aumento do custo em torno de 4,86% e uma perda considerável do valor em 6,5%. Também nessa amostra, a solução 31 também apresenta uma boa taxa de redução de risco (10,6%), mas da mesma forma que a solução 35, essa redução do risco é obtida com um aumento do custo (3,20%) e uma perda de valor significativa próximo a 6,19%.

Diversas outras soluções apresentam redução de risco acima de 9,5%, algumas até apresentando também redução do custo, porém mantendo a mesma perspectiva de perda acentuada do valor. Como exemplo dessa situação, a solução 15 mostra redução de risco em 9,60%, redução do custo em 0,30%, mas perda de valor em cerca de 6,45%.

O próximo histograma (Figura B.11) apresenta os resultados para o mesmo ROI e *dataset*, porém com o custo sem limitações no espaço de busca e o valor sujeito à restrição imposta pela margem de tolerância. Nessa Figura, uma amostra de 30 das 375 soluções obtidas são apresentadas. A vizinhança para essa conjuntura está em torno da solução 8 (solução inicial).



Figura B.11: Histograma de variações percentuais para a região de interesse 2 (*dataset-2*), com custo livre

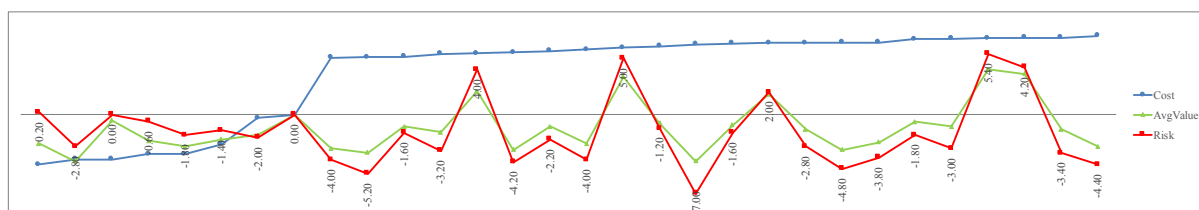


Figura B.12: Tendência de variações para a região de interesse 2 (*dataset-2*), com custo livre

Nessa amostra, a solução 19 apresenta uma redução de risco próximo a 7,00%, mas apresentando um aumento de custo em 6,24% e uma perda de valor de 4,13%. No geral, esse setor não apresenta grandes reduções de risco. Esse comportamento pode ser também identificado no gráfico de tendências anterior (Figura B.12).

Por fim, os resultados para o *dataset-2* no que diz respeito à Região de Interesse 3 são apresentados a seguir. Inicialmente, as Figuras (B.13 e B.14) apresentam os dados quando o valor não está restrito a nenhuma margem de tolerância.

No histograma acima, é possível identificar que há soluções com redução de risco acima de 10,00%, como por exemplo, as soluções 26 e 17. No caso da primeira, há uma redução do risco em 10,53% e também uma redução do custo próximo a 1,11%, no entanto, a perda de valor é acentuada (8,62%). No caso da solução 17, a redução do risco é de 10,24%

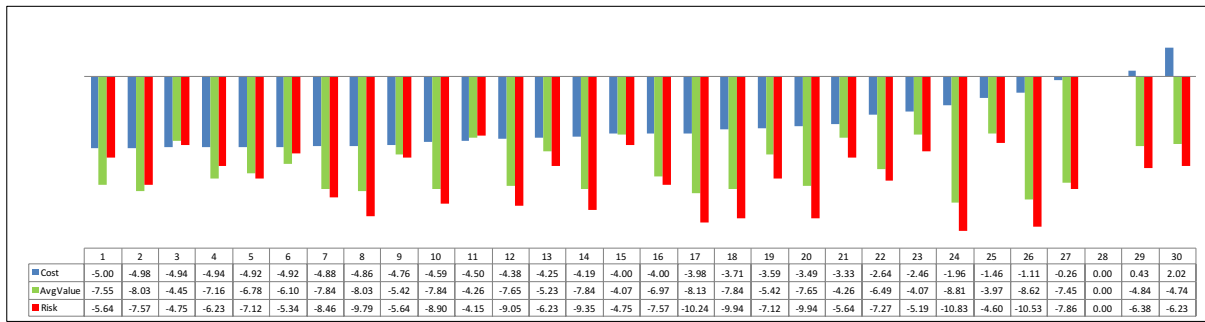


Figura B.13: Histograma de variações percentuais para a região de interesse 3 (*dataset-2*), com valor livre

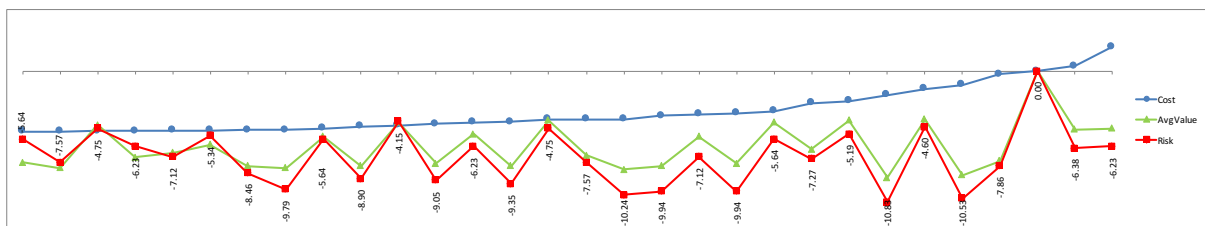


Figura B.14: Tendência de variações para a região de interesse 3 (*dataset-2*), com valor livre

acompanhada de um decréscimo do custo em torno de 3,98%, mas com uma perda de valor significativa próxima a 8,13%. Algumas outras soluções (soluções 8 e 18) apresentam reduções de risco um pouco inferior a 10,00% (9,79% e 9,94%, respectivamente), mas também apresentando perdas de valores significativas próximas a 8,00%. No gráfico de tendências (Figura B.14) é possível visualizar o comportamento de perda e ganhos para os objetivos e identificar os picos de maiores reduções de risco. As duas próximas Figuras, mostram os resultados para o mesmo *dataset-2* e ROI3, porém quando o custo está livre para variar e o valor está restrito à margem de tolerância.

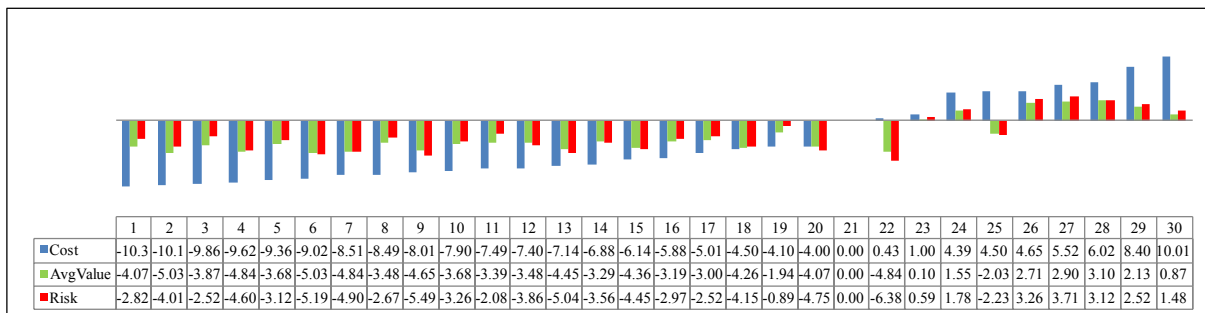


Figura B.15: Histograma de variações percentuais para a região de interesse 3 (*dataset-2*), com custo livre

A Figura B.15 apresenta uma amostra contendo 30 soluções das 53 possíveis. Nesse caso é possível identificar a solução 22, que possui uma redução de custo próximo a

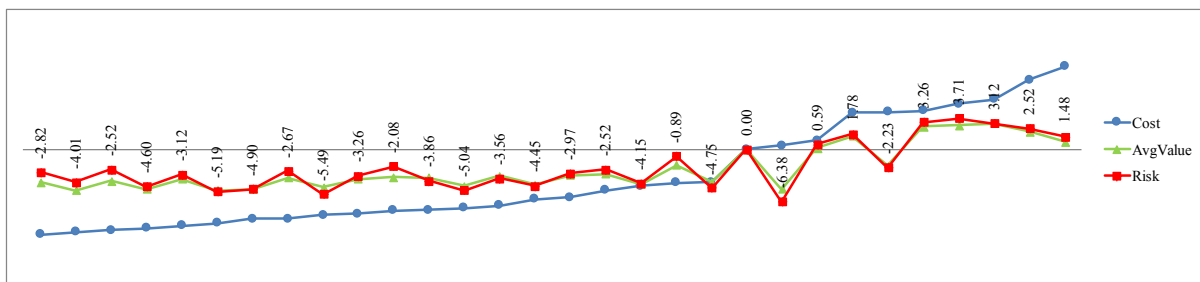


Figura B.16: Tendência de variações para a região de interesse 3 (*dataset-2*), com custo livre

6,38%, um aumento pequeno no custo em torno de 0,43% e uma redução do valor em cerca de 4,84%. Nesse cenário, algumas soluções possuem baixa redução do risco, porém um decréscimo acentuado do custo, tais como, as soluções 1 e 2, em que apresentam diminuição do custo em 10,3% e 10,1%, respectivamente. Essa característica pode ser bem visualizada no gráfico de tendências (Figura B.16), no qual é possível perceber que na relação perda/ganho dos objetivos é possível perceber que a linha referente ao custo, em boa parte do gráfico, está abaixo da linha com as soluções (pontos) associados ao risco.

RESULTADOS ADICIONAIS PARA O TRATAMENTO 3

Os mesmos dados (Ponto P para cada ROI) para os experimentos no tratamento 1 foram utilizados (ver Tabelas 6.9 e 6.10). Os diagramas a seguir mostram os resultados para o ROI 1 do *dataset-1* (Figura C.1 e Figure C.2).

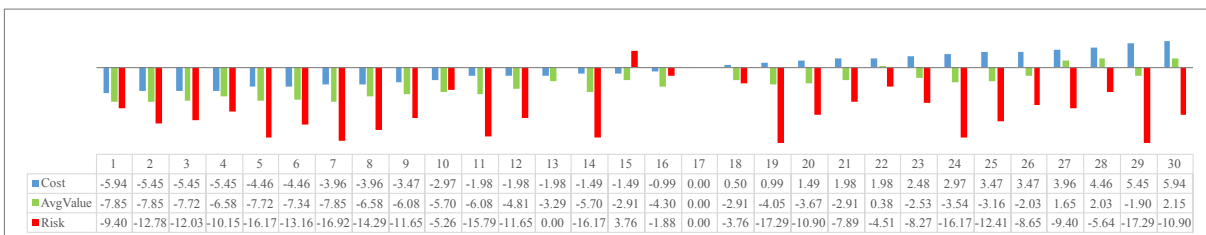


Figura C.1: Histograma de variações de percentagens para o ROI 1 (*dataset-1*) - Tratamento 3

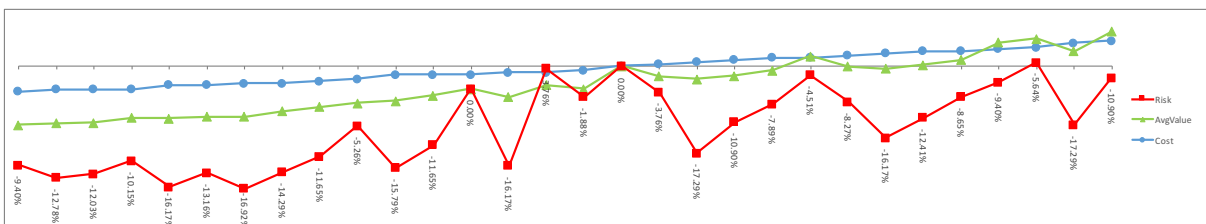


Figura C.2: Tendência de variações para o ROI 1 (*dataset-1*)- Tratamento 3

Para esse cenário experimental, uma amostra de 30 soluções das 65 possíveis é apresentada. A vizinhança das soluções RA-MONRP está em torno da solução 17, escolhida como ponto de referência. A solução 19 possui um alto índice de redução do risco (17,29%) acompanhado de um acréscimo pequeno do custo em torno de 0,99% e uma perda de valor próxima a 4,05%. A solução 14 também apresenta uma percentagem alta de redução do risco (16,17%) e, diferentemente da citada acima, mostra uma redução do custo da

solução em torno de 1,49%, porém, dessa vez com uma perda de valor um pouco maior (5,70%). De forma similar, a solução 12 apresenta uma redução de custo em 1,98%, acompanhado de uma perda de valor em torno de 4,81%, e com uma redução de risco considerável próximo a 11,65%. Esse cenário apresenta diversas soluções com redução do risco acima de 10,15%, porém quase sempre acompanhada de uma perda considerável do valor (ex: Soluções 3 a 9). O gráfico de tendências a seguir apresenta o comportamento conjunto da tripla custo, valor e risco.

No que diz respeito aos resultados da Região de Interesse 3 para o *dataset-1*, a Figura C.3 mostra o histograma desse cenário e a Figura C.4 mostra o gráfico de tendência das soluções.

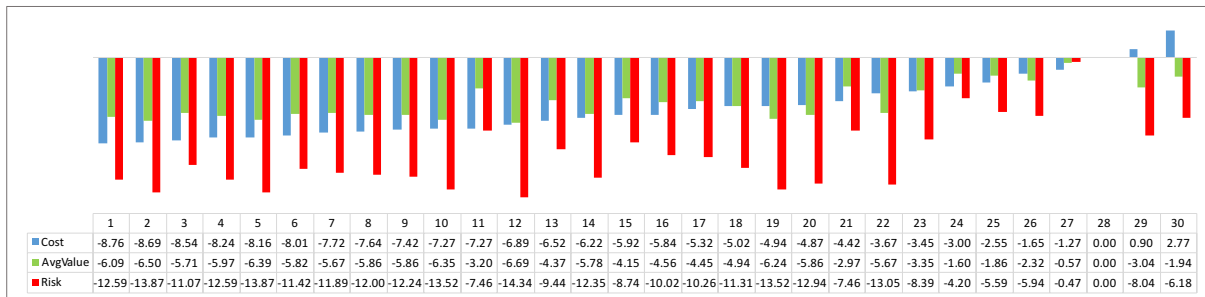


Figura C.3: Histogramas de variações de percentagens para o ROI 3 (*dataset-1*) - Tratamento 3

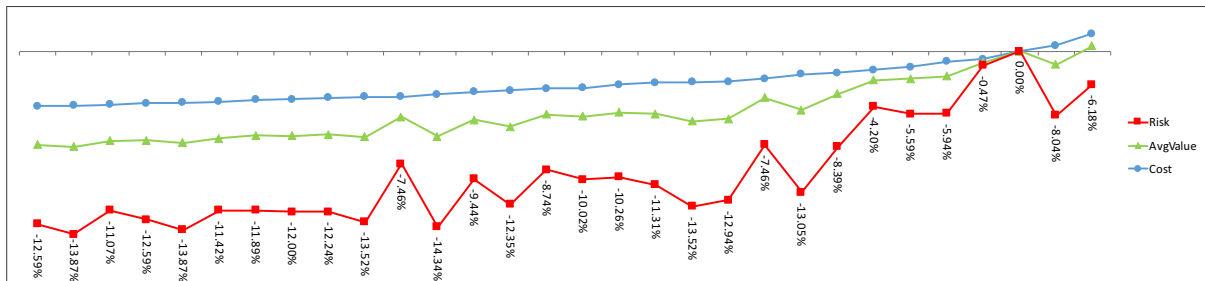


Figura C.4: Tendência de variações para o ROI 3 (*dataset-1*) - Tratamento 3

Nesse caso, é possível notar que algumas soluções apresentam alta redução do risco, com uma acentuada diminuição de custos e alguma perda menor de valor. A título de exemplo, a solução 8 tem um risco de insatisfação redução de cerca de 12%, uma redução de custo próximo de 7,64% e uma perda de valor em 5,86%.

A solução 2, por exemplo, possui uma queda acentuada do risco (13,87%) e também um custo menor em relação à solução P inicial (Solução 28), porém, apresenta também uma redução em cerca de 6,50% do valor. É bastante evidente no gráfico de tendência (ver Figura C.4) que uma redução de risco elevado é possível, embora seja associada a um correspondente, embora menor, perda de valor.

Em relação aos resultados para o segundo conjunto de dados (*dataset-2*), as Figuras C.5 e C.6 a seguir mostram os resultados para a Região de Interesse 1, no caso do tratamento 3.

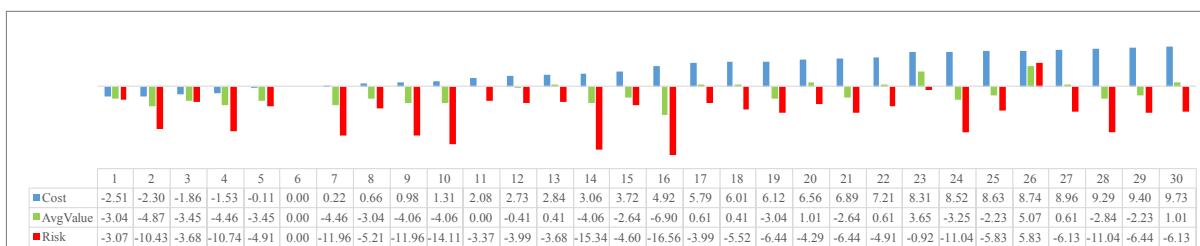


Figura C.5: Histogramas de variações de percentagens para o ROI 1 (*dataset-2*) - Tratamento 3

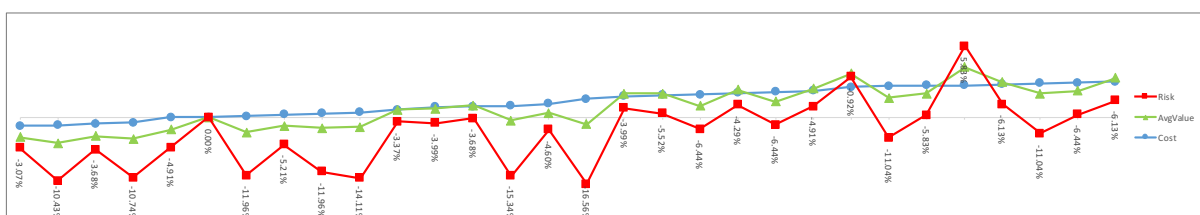


Figura C.6: Tendência de variações para o ROI 1 (*dataset-2*) - Tratamento 3

A partir do gráfico de tendências é possível identificar as soluções que possuem maior pico de redução do risco. Tanto nesse gráfico como no histograma uma amostra de 30 das 58 soluções são apresentadas.

Nesse cenário, por exemplo, é possível identificar a solução 16, que apresenta uma redução do risco em relação à solução inicial (solução 6) próximo a 16,56%, porém com uma perda de valor em torno de 6,90% e com um aumento do custo em cerca de 4,92%. Outra situação com bom índice de redução do risco pode ser visto na solução 7, onde o risco apresenta uma taxa percentual de redução no valor de 11,96%, mas dessa vez, com uma perda um pouco menor do valor (4,96%) em relação à solução citada anteriormente e com um aumento de custo pequeno em torno de 0,22%. De uma forma um pouco distinta, a solução 2 apresenta ao invés de um aumento do custo, uma redução razoável desse objetivo (2,51%). Mas mantém um bom índice de redução do risco próximo a 10,43% e uma perda de valor abaixo de 5%. A solução 4 mantém a mesma configuração da solução 2, ou seja, diminuição do custo (1,53%), perda de valor (4,87%) e boa taxa de redução do risco (10,74%).

Por fim, os últimos resultados do tratamento 3 que dizem respeito à terceira Região de Interesse. Nesse caso, as medidas de percentagem das soluções da vizinhança são comparadas com a solução 19, escolhida para ser a solução de referência.

Algumas soluções apresentam boas taxas de redução de risco, porém também com perda de valor acentuadas. Por exemplo, a solução 1 obteve uma redução de risco próximo a 10,3%, com uma redução de custo em cerca de 6,00%, porém com uma perda de valor em torno de 9,20%. Da mesma forma, a solução 12 apresenta uma redução do risco em cerca de 11,4%, com um decréscimo do custo próximo a 4,83% e um perda significativa de valor em torno de 9,49%. A solução 14 também é um exemplo similar, em que se tem redução de risco (9,05%), diminuição do custo (4,25%) e perda significativa em torno

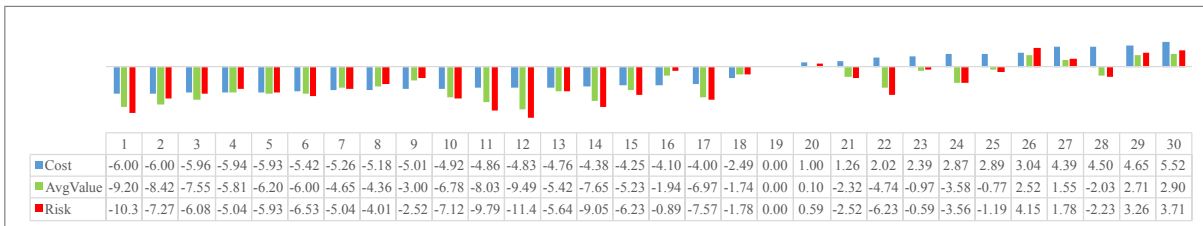


Figura C.7: Histogramas de variações de percentagens para o ROI 3 (*dataset-2*) - Tratamento 3

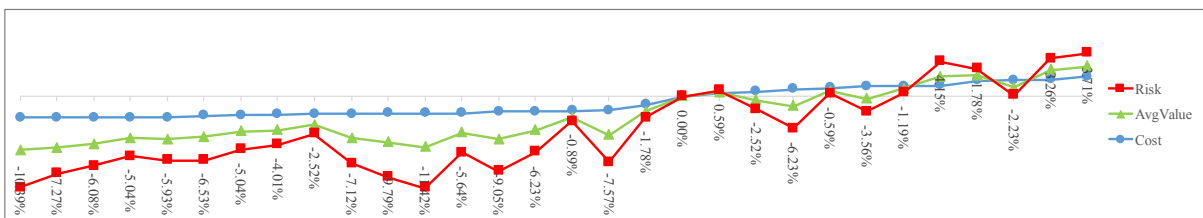


Figura C.8: Tendência de variações para o ROI 3 (*dataset-2*) - Tratamento 3

de 7,65%. O comportamento da relação perda/ganho dos objetivos para essa Região de Interesse pode ser visto no gráfico de tendências (Figura C.8).

REFERÊNCIAS BIBLIOGRÁFICAS

- AASEM, M. et al. Analysis and optimization of software requirements prioritization techniques. *Information and Emerging Technologies (ICIET)*, 2010, 2010.
- ABRAHAM, A.; JAIN, L. C.; GOLDBERG, R. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN 1852337877.
- AGARWAL, N.; RATHOD, U. Defining ‘success’ for software projects: An exploratory revelation. *International Journal of Project Management*, v. 24, n. 4, p. 358–370, may 2006. ISSN 02637863.
- AHUJA, R. K.; MOHRING, R. H.; ZAROLIAGIS, C. D. (Ed.). *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*. Springer, 2009. (Lecture Notes in Computer Science, v. 5868). ISBN 978-3-642-05464-8. Disponível em: <http://dx.doi.org/10.1007/978-3-642-05465-5>.
- AKKER, J. M. V. D. et al. Determination of the Next Release of a Software Product : an Approach using Integer Linear. *Proceedings of CAISE’05*, v. 03018, p. 119–124, 2005.
- AKKER, J. M. V. D. et al. Flexible Release Planning using Integer Linear Programming. In: *Proceedings of the 11th International Workshop on Requirements Engineering for Software Quality (RefsQ ’05)*. [S.l.: s.n.], 2005.
- AKKER, M. van den et al. Software product release planning through optimization and what-if analysis. *Information and Software Technology*, v. 50, n. 1-2, p. 101–111, jan 2008. ISSN 09505849.
- AMBER, S.; SHAWOO, N.; BEGUM, S. Determination of risk during requirement engineering process. *Journal of Emerging Trends in Computing and Information Sciences*, v. 3, n. 3, p. 358–364, 2012. ISSN 22186301.
- ARCURI, A.; BRIAND, L. A practical guide for using statistical tests to assess randomized algorithms in software engineering. *33rd International Conference on Software Engineering (ICSE)*, p. 1–10, 2011. ISSN 0270-5257.
- ASNAR, Y.; GIORGINI, P.; MYLOPOULOS, J. Goal-driven risk assessment in requirements engineering. *Requirements Engineering*, v. 16, p. 101–116, 2011. ISSN 09473602.
- AZMEH, Z.; MIRBEL, I.; CRESCENZO, P. Highlighting stakeholder communities to support requirements decision-making. In: _____. *Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen*,

Germany, April 8-11, 2013. *Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 190–205. ISBN 978-3-642-37422-7.

BAGNALL, A.; RAYWARD-SMITH, V.; WHITTLEY, I. The next release problem. *Information and Software Technology*, v. 43, n. 14, p. 883–890, dez. 2001. ISSN 09505849. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S095058490100194X>.

BAKER, P.; HARMAN, M.; STEINH, K. Search Based Approaches to Component Selection and Prioritization for the Next Release Problem. *Conference On Software Maintenance*, 2006.

BANNERMAN, P. L. Risk and risk management in software projects: A reassessment. *Journal of Systems and Software*, v. 81, n. 12, p. 2118–2133, 2008. ISSN 01641212.

BARROS, M. D. O.; DIAS-NETO, A. C. Desenvolvendo uma Abordagem Sistemática para Avaliação dos Estudos Experimentais em Search-Based Software Engineering. In: *II Workshop de Engenharia de Software baseada em Buscas. Congresso Brasileiro de Software: teoria e prática*. [S.l.: s.n.], 2011.

BASILI, V. R. Experimental software engineering issues: Critical assessment and future directions: International workshop dagstuhl castle, germany, september 14–18, 1992 proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. cap. The experimental paradigm in software engineering, p. 1–12. ISBN 978-3-540-47903-1.

BECHIKH, S.; SAID, L. B.; GHÉDIRA, K. Group Preference-based Evolutionary Multi-objective Optimization with Non-Equally Important Decision Makers : Application to the Portfolio Selection Problem. v. 5, n. 2012, p. 278–288, 2013.

BERANDER, P.; ANDREWS, A. Engineering and managing software requirements. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. cap. Requirements Prioritization, p. 69–94. ISBN 978-3-540-28244-0.

BEYER, H.-G.; SENDHOFF, B. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, v. 196, p. 3190 – 3218, 2007. ISSN 0045-7825.

BIERNACKI, P.; WALDORF, D. Snowball sampling: Problems and techniques of chain referral sampling. *Sociological Methods and Research*, v. 10, n. 2, p. 141–163, 1981.

BOEHM, B.; GRÜNBACHER, P.; BRIGGS, R. O. Developing groupware for requirements negotiation: Lessons learned. *IEEE Softw.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 18, n. 3, p. 46–55, maio 2001. ISSN 0740-7459.

BRANKE, J. et al. (Ed.). *Multiobjective Optimization, Interactive and Evolutionary Approaches [outcome of Dagstuhl seminars]*, v. 5252 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 5252). [S.l.]: Springer, 2008. ISBN 978-3-540-88907-6.

- BRASIL, M. et al. A Multiobjective Optimization Approach to the Software Release Planning with Undefined Number of Releases and Interdependent Requirements. *Enterprise Information Systems*, 2012.
- CAI, X.; WEI, O. A hybrid of decomposition and domination based evolutionary algorithm for multi-objective software next release problem. *2013 10th IEEE International Conference on Control and Automation (ICCA)*, Ieee, p. 412–417, jun. 2013. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6565143>.
- CAI, X.; WEI, O.; HUANG, Z. Evolutionary approaches for multi-objective next release problem. *Computing and Informatics*, v. 31, n. 4, p. 847, 2012.
- CARLSHAMRE, P. et al. An industrial survey of requirements interdependencies in software product release planning. In: *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*. [S.l.: s.n.], 2001. p. 84–91.
- CHENG, B.; ATLEE, J. Research directions in requirements engineering. In: *Future of Software Engineering, 2007. FOSE '07*. [S.l.: s.n.], 2007. p. 285–303.
- CLARKE, J. et al. Reformulating software engineering as a search problem. *IEEE Proceedings - Software*, v. 150, n. 3, p. 161, 2003. ISSN 14625970.
- COELLO, C. A. Handling preferences in evolutionary multiobjective optimization: A survey. In: *In 2000 Congress on Evolutionary Computation*. [S.l.: s.n.], 2000. p. 30–37.
- COELLO, C. C.; LAMONT, G.; VELDHUIZEN, D. van. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd. ed. [S.l.]: Springer, 2007. (Genetic and Evolutionary Computation).
- COLARES, F. et al. A New Approach to the Software Release Planning. *2009 XXIII Brazilian Symposium on Software Engineering*, Ieee, p. 207–215, out. 2009. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5336083>.
- De Moura, L.; RNER, N. B. Satisfiability modulo theories: An appetizer. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 5902 LNCS, p. 23–36, 2009. ISSN 03029743.
- DEB, K.; KALYANMOY, D. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, IEEE Press, Piscataway, NJ, USA, v. 6, n. 2, p. 182–197, abr. 2002. ISSN 1089-778X.
- DEB, K. et al. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. *International Journal of Computational Intelligence Research*, v. 2, n. 3, p. 273–286, 2006. ISSN 09741259.

DURILLO, J. J. et al. A Study of the Multi-objective Next Release Problem. *2009 1st International Symposium on Search Based Software Engineering*, Ieee, p. 49–58, maio 2009. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5033180>.

DURILLO, J. J. et al. *A study of the bi-objective next release problem*. [S.l.: s.n.], 2011. 29–60 p. ISSN 1382-3256. ISBN 1066401091473.

DUTERTRE, B.; MOURA, L. de. *The Yices SMT solver*. [S.l.], 2006.

DYBA, T.; DINGSOYR, T.; HANSSEN, G. K. Applying Systematic Reviews to Diverse Study Types: An Experience Report. *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Ieee, n. 7465, p. 225–234, sep 2007.

EHRGOTT, M. *Multicriteria optimization*. [S.l.]: Springer-Verlag, 2000. (Lecture Notes in Economics and Mathematical Systems).

FAIRLEY, R.; BOURQUE, P. (Ed.). *Guide to the Software Engineering Body of Knowledge (SWEBOOK) , Version 3.0*. [S.l.: s.n.], 2014. ISBN 9780769551661.

FAZIO, R. H. et al. Variability in automatic activation as an unobtrusive measure of racial attitudes: A bona fide pipeline? *Journal of Personality and Social Psychology*, v. 69, 1995.

FEATHER, M. S.; CORNFORD, S. L. Quantitative risk-based requirements reasoning. *Requirements Engineering*, v. 8, p. 248–265, 2003. ISSN 0947-3602.

FELLOWS, L.; HOOKS, I. A case for priority classifying requirements. *INCOSE International Symposium*, v. 8, n. 1, p. 747–753, 1998. ISSN 2334-5837.

FINKELSTEIN, A. et al. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements Engineering*, v. 14, n. 4, p. 231–245, 2009. ISSN 0947-3602.

G. Freitas, F.; P. Coutinho, D.; T. Souza, J. Software Next Release Planning Approach through Exact Optimization. *International Journal of Computer Applications*, v. 22, n. 8, p. 1–8, may 2011. ISSN 09758887.

GENDREAU, M.; POTVIN, J.-Y. Search methodologies: Introductory tutorials in optimization and decision support techniques. In: _____. Boston, MA: Springer US, 2005. cap. Tabu Search, p. 165–186. ISBN 978-0-387-28356-2.

GENDREAU, M.; POTVIN, J.-Y. *Handbook of Metaheuristics*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 1441916636, 9781441916631.

GREER, D.; RUHE, G. Software release planning: an evolutionary and iterative approach. *Information and Software Technology*, v. 46, n. 4, p. 243–253, mar 2004. ISSN 09505849.

GROUP, T. S. *The CHAOS Report*. [S.l.], 2014.

GUEORGUIEV, S.; HARMAN, M.; ANTONIOL, G. Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. (GECCO '09), p. 1673–1680. ISBN 978-1-60558-325-9. Disponível em: <http://doi.acm.org/10.1145/1569901.1570125>.

HAN, W. M.; HUANG, S. J. An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software*, v. 80, n. 1, p. 42–50, 2007. ISSN 01641212.

HARMAN, M. The current state and future of search based software engineering. In: *2007 Future of Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2007. (FOSE '07), p. 342–357. ISBN 0-7695-2829-5.

HARMAN, M.; JONES, B. F. Search-based software engineering. *Information and Software Technology*, v. 43, n. 14, p. 833–839, dec 2001. ISSN 09505849.

HARMAN, M. et al. Search based data sensitivity analysis applied to requirement engineering. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. (GECCO '09), p. 1681–1688. ISBN 978-1-60558-325-9.

HARMAN, M. et al. Search Based Software Engineering : Techniques, Taxonomy, Tutorial. In: NORDIO, B. M.; MARTIN (Ed.). *Empirical Software Engineering and Verification*. [S.l.]: Springer, 2012. p. 1–59.

HEGER, D. A Disquisition on the Performance Behavior of Binary Search Tree Data Structures. *Journal on Software Process Technology*, oct 2004.

HILLIER, F. S.; LIEBERMAN, G. J. *Introduction to Operations Research*. Seventh. New York, USA: McGraw-Hill, 2004.

HOODAT, H.; RASHIDI, H. Classification and Analysis of Risks in Software Engineering. *International Journal of Computer, Electrical, Automation, Control and Information Engineering.*, v. 3, n. 8, p. 446–452, 2009. ISSN 2010376X. Disponível em: <http://www.waset.ac.nz/journals/waset/v56/v56-83.pdf>.

HULL, M. E. C.; JACKSON, K.; DICK, J. *Requirements Engineering, Second Edition*. [S.l.]: Springer, 2005. ISBN 978-1-85233-879-4.

IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. [S.l.], 1990. v. 121990, n. 1, 1 p. Disponível em: <http://ieeexplore.ieee.org/xpls/abs{\\-}all.jsp?arnumber=159>.

IEEE98. Ieee recommended practice for software requirements specifications. *IEEE Std 830-1998*, p. 1–40, Oct 1998.

IORDACHE, R.; IORDACHE, S.; MOLDOVEANU, F. A framework for the study of preference incorporation in multiobjective evolutionary algorithms. *Proceedings of the 2014 conference on Genetic and evolutionary computation - GECCO '14*, ACM Press, New York, New York, USA, p. 621–628, 2014. Disponível em: <http://dl.acm.org/citation.cfm?doid=2576768.2598380>.

ISLAM, S.; HOUMB, S. H. Integrating risk management activities into requirements engineering. *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)*, p. 299–310, 2010. ISSN 2151-1349. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5507389>.

JEDLITSCHKA, A.; PFAHL, D. Reporting guidelines for controlled experiments in software engineering. In: *2005 International Symposium on Empirical Software Engineering (ISESE, 2005), 17-18 November 2005, Noosa Heads, Australia*. [s.n.], 2005. p. 95–104. Disponível em: <http://dx.doi.org/10.1109/ISESE.2005.1541818>.

JIANG, H.; XUAN, J.; REN, Z. Approximate backbone based multilevel algorithm for next release problem. *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*, ACM Press, New York, New York, USA, p. 1333, 2010.

JIANG, H. et al. A Hybrid ACO Algorithm for the Next Release Problem. *Proceedings of 2nd International Conference on Software Engineering and Data Mining (SEDM 2010)*, n. 60805024, p. 166–171, 2010.

JORGENSEN, M. Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Information and Software Technology*, Elsevier B.V., v. 52, n. 5, p. 506–516, 2010. ISSN 09505849. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2009.12.002>.

KANDT, R. *Software Requirements Engineering: Practices and Techniques*. California, 2003. 35 p.

KARIM, M.; RUHE, G. Bi-objective genetic search for release planning in support of themes. In: GOUES, C. L.; YOO, S. (Ed.). *Search-Based Software Engineering*. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8636). p. 123–137. ISBN 978-3-319-09939-2. Disponível em: http://dx.doi.org/10.1007/978-3-319-09940-8_9.

KARLSSON, J.; RYAN, K. A cost-value approach for prioritizing requirements. *IEEE Softw.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 14, n. 5, p. 67–74, set. 1997. ISSN 0740-7459. Disponível em: <http://dx.doi.org/10.1109/52.605933>.

- KARLSSONA, J. et al. An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, v. 39, n. 14-15, p. 939–947, jan 1998. ISSN 09505849.
- KÖCHE, J. C. *Fundamentos de metodologia científica: teoria da ciência e iniciação à pesquisa*. 29th. ed. [S.l.]: Vozes, 2011. ISBN 978-85-32618047.
- KEIL, M. et al. A framework for identifying software project risks. *Communications of the ACM*, v. 41, n. 11, p. 83, 1998. ISSN 00010782.
- KHEIRKHAH, E.; DERAMAN, A. Important factors in selecting Requirements Engineering Techniques Esmaeil Kheirkhah. *IEEE*, 2008.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. ISSN 0036-8075.
- KITCHENHAM, B. *Procedures for performing systematic reviews*. [S.l.], 2004.
- KITCHENHAM, B. et al. Can we evaluate the quality of software engineering experiments? In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. New York, NY, USA: ACM, 2010. (ESEM '10), p. 2:1–2:8. ISBN 978-1-4503-0039-1.
- KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering: Processes and Techniques*. 1st. ed. [S.l.]: Wiley Publishing, 1998. ISBN 0471972088, 9780471972082.
- KUKREJA, N. et al. Selecting an appropriate framework for value-based requirements prioritization. In: *Requirements Engineering Conference (RE), 2012 20th IEEE International*. [S.l.: s.n.], 2012. p. 303–308. ISSN 1090-750X.
- KUMARI, a. C.; SRINIVAS, K.; GUPTA, M. P. Software requirements selection using Quantum-inspired Multi-objective Differential Evolution Algorithm. *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, Ieee, p. 1–8, sep 2012.
- LAKATOS, M.; MARCONI, E. *Fundamentos de metodologia científica*. 5th. ed. [S.l.]: Atlas, 2003. ISBN 8522433976.
- LAMSWEERDE, A. van. *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley, 2009. ISBN 978-0-470-01270-3. Disponível em: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-EHEP000863.html>.
- LETIER, E.; STEFAN, D.; BARR, E. T. Uncertainty, Risk, and Information Value in Software Requirements and Architecture. *Proceedings of the 36th International Conference on Software Engineering*, p. 883–894, 2014. ISSN 15581225.

LI, C. et al. Requirements engineering: Foundation for software quality: 13th international working conference, refsq 2007, trondheim, norway, june 11-12, 2007. proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. cap. Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product, p. 93–108. ISBN 978-3-540-73031-6.

Li, C. et al. An integrated approach for requirement selection and scheduling in software release planning. *Requirements engineering*, Springer, Berlin, Germany, v. 15, n. 4, p. 375–396, November 2010.

LI, L.; HARMAN, M.; LETIER, E. Robust Next Release Problem : Handling Uncertainty During Optimization. *Proceedings of 14th Annual onference on Genetic and evolutionary computation - GECCO '14, July 12-16, Vancouver, 2014*, p. 1247–1254, 2014.

LI, X.; XIE, M.; NG, S. H. Multi-Objective Optimization Approaches To Software Release Time Determination. *Asia-Pacific Journal of Operational Research*, v. 29, n. 03, p. 1240019, jun 2012. ISSN 0217-5959.

LIKERT, R. A technique for the measurement of attitudes. *Archives of Psychology*, v. 22, n. 140, p. 1–55, 1932.

LIMA, D. C. et al. Search based software engineering: Third international symposium, ssbse 2011, szeged, hungary, september 10-12, 2011. proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. cap. A Fuzzy Approach to Requirements Prioritization, p. 64–69. ISBN 978-3-642-23716-4.

LIMA, M. M. *The book of trees : visualizing branches of knowledge*. New York: Princeton Architectural Press, [s.n.], 2014.

LINZ, S.; JOHN, K. S.; SEMPLE, C. Optimizing tree and character compatibility across several phylogenetic trees. *Theoretical Computer Science*, v. 513, p. 129 – 136, 2013. ISSN 0304-3975.

LOUCOPOULOS, P.; KARAKOSTAS, V. *System Requirements Engineering*. New York, NY, USA: McGraw-Hill, Inc., 1995. ISBN 0077078438.

MAUŠA, G. et al. Hill climbing and simulated annealing in large scale next release problem. In: *EUROCON, 2013 IEEE*. [S.l.: s.n.], 2013. p. 452–459.

MCNEIL, A. J.; FREY, R.; EMBRECHTS, P. *Quantitative risk management : concepts, techniques and tools*. Princeton (N.J.): Princeton University Press, 2005. (Princeton series in finance). ISBN 0-691-12255-5. Disponível em: <<http://opac.inria.fr/record=b1132295>>.

MELI, R. SAFE: a method to understand, reduce, and accept project risk. In: *ESCOM-ENCRESS 98 - Project Control for 2000 and Beyond*. Rome-Italy: [s.n.], 1998.

- MELI, R. Risks , requirements and estimation of a software project. In: *proceedings of ESCOM - SCOPE 99, 27-29 April 1999, Herstmonceux Castle, United Kingdom : combining the 10th European Software Control and Metrics conference with the 2nd Software Certification PrOgramme in Europe Conference*. [S.l.: s.n.], 1999. p. 1–14.
- MOORES, T.; CHAMPION, R. A methodology for measuring the risk associated with a software requirements specification. *Australasian Journal of Information Systems*, v. 4, n. 1, 1996.
- MOURA, L. de; BJØRNER, N. Z3: An efficient smt solver. In: *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 4963), p. 337–340.
- NGO-THE, A.; RUHE, G. A systematic approach for solving the wicked problem of software release planning. *Soft Computing*, v. 12, n. 1, p. 95–108, jun. 2008. ISSN 1432-7643. Disponível em: <http://www.springerlink.com/index/10.1007/s00500-007-0219-2>.
- OCHOA, G. Search methodologies in real-world software engineering. In: *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2013. (GECCO '13 Companion), p. 1085–1088. ISBN 978-1-4503-1964-5. Disponível em: <http://doi.acm.org/10.1145/2464576.2482687>.
- ORTEGA, F. et al. Incorporating group recommendations to recommender systems: Alternatives and performance. *Information Processing & Management*, Elsevier Ltd, v. 49, n. 4, p. 895–901, jul. 2013. ISSN 03064573. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0306457313000174>.
- PAIXAO, M.; SOUZA, J. A scenario-based robust model for the next release problem. *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13*, ACM Press, New York, New York, USA, p. 1469, 2013.
- PAIXAO, M. H. E.; SOUZA, J. T. de. A recoverable robust approach for the next release problem. In: RUHE, G.; ZHANG, Y. (Ed.). *SSBSE*. [S.l.]: Springer, 2013. (Lecture Notes in Computer Science, v. 8084), p. 172–187. ISBN 978-3-642-39741-7.
- Peter Goos, D. M. *Statistics with JMP: Graphs, Descriptive Statistics and Probability*. [S.l.]: Wiley; 1 edition (May 4, 2015), 2015. 368 p. ISBN 978-1-119-03570-1.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swinton, UK, UK: British Computer Society, 2008. (EASE'08), p. 68–77.
- PFLEEGER, S. *Engenharia de Software - Teoria e Prática*. 2ed.. ed. [S.l.]: Prentice Hall do Brasil, 2004.

PITANGUEIRA, A. M.; MACIEL, R. S. P.; BARROS, M. Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. *Journal of Systems and Software*, v. 103, n. 0, p. 267 – 280, 2015. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121214002118>.

PITANGUEIRA, A. M. et al. Search based software engineering: 5th international symposium, ssbse 2013, st. petersburg, russia, august 24-26, 2013. proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. cap. A Systematic Review of Software Requirements Selection and Prioritization Using SBSE Approaches, p. 188–208. ISBN 978-3-642-39742-4.

PITANGUEIRA, A. M. et al. Risk-Aware Multi-stakeholder Next Release Planning Using Multi-objective Optimization. In: PASTOR, M. D.; O. (Ed.). *22nd International Working Conference on Requirements Engineering: Foundation for Software Quality*. [S.l.]: Springer International Publishing Switzerland LNCS 9619, 2016. p. 1–15.

POHL, K. The three dimensions of requirements engineering. *Advanced Information Systems Engineering*, v. 6353, p. 275–292, 1993. ISSN 03064379.

POHL, K. *Requirements Engineering: Fundamentals, Principles, and Techniques*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642125778, 9783642125775.

POHL, K.; RUPP, C. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant*. 1st. ed. [S.l.]: Rocky Nook, 2011. ISBN 1933952814, 9781933952819.

POLI, R.; LANGDON, W. B.; MCPHEE, N. F. *A Field Guide to Genetic Programming*. [S.l.]: Lulu Enterprises, UK Ltd, 2008. ISBN 1409200736, 9781409200734.

PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 5th. ed. [S.l.]: McGraw-Hill Higher Education, 2001. ISBN 0072496681.

PURSHOUSE, R. C. et al. *A Review of Hybrid Evolutionary Multiple Criteria Decision Making Methods*. [S.l.], 2014. 1–8 p.

RANA, A. et al. Categorization of Software Release Risks and Its Abatement Strategy. *Journal of Software Engineering and Applications*, v. 7, n. November, p. 1039–1044, 2014.

REGNELL, B.; KUHCINSKI, K. Exploring Software Product Management decision problems with constraint solving - opportunities for prioritization and release planning. *2011 Fifth International Workshop on Software Product Management (IWSPM)*, Ieee, p. 47–56, ago. 2011. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6046203>.

ROBERTSON, S.; ROBERTSON, J. *Mastering the Requirements Process*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999. ISBN 0-201-36046-2.

RONG, J. et al. An approach to measuring software development risk based on information entropy. In: *Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on*. [S.l.: s.n.], 2009. v. 2, p. 296–298.

RUHE, G.; TN, A. B.; GREER, D. Quantitative Studies in Software Release Planning under Risk and Resource Constraints University of Calgary. *Empirical Software Engineering*, p. 1–10, 2003.

SAATY, T. *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. [S.l.]: McGraw-Hill, New York,, 1980.

SAGRADO, J. D. et al. Requirements selection: Knowledge based optimization techniques for solving the next release problem. In: *6th Workshop on Knowledge Engineering and Software Engineering (KESE 2010)*. [S.l.: s.n.], 2010. ISSN 16130073.

SAGRADO, J. D.; Del Aguila, I. M.; ORELLANA, F. J. Ant Colony Optimization for the Next Release Problem: A Comparative Study. *2nd International Symposium on Search Based Software Engineering*, Ieee, p. 67–76, sep 2010.

SALIU, M. O.; RUHE, G. Software release planning for evolving systems. *ISSE*, v. 1, n. 2, p. 189–204, 2005.

SALIU, M. O.; RUHE, G. Supporting software release planning decisions for evolving systems. In: *SEW*. [S.l.]: IEEE Computer Society, 2005. p. 14–26.

SALIU, M. O.; RUHE, G. Bi-objective release planning for evolving software systems. In: *ESEC/SIGSOFT FSE*. [S.l.]: ACM, 2007. p. 105–114.

SAYYAD, A. S.; MENZIES, T.; AMMAR, H. On the value of user preferences in search-based software engineering: A case study in software product lines. *2013 35th International Conference on Software Engineering (ICSE)*, Ieee, p. 492–501, may 2013.

SIM, W. W.; BROUSE, P. S. Empowering Requirements Engineering Activities with Personas. *Procedia Computer Science*, v. 28, p. 237–246, 2014. ISSN 18770509.

SJOBERG, D. I. K. et al. A survey of controlled experiments in software engineering. *IEEE Trans. Softw. Eng.*, IEEE Press, Piscataway, NJ, USA, v. 31, n. 9, p. 733–753, set. 2005. ISSN 0098-5589.

SOMMERVILLE, I. *Software Engineering (7th Edition)*. [S.l.]: Pearson Addison Wesley, 2004. ISBN 0321210263.

SOUZA, J. T. de et al. An ant colony optimization approach to the software release planning with dependent requirements. In: *Proceedings of the Third International Conference on Search Based Software Engineering*. Berlin, Heidelberg: Springer-Verlag, 2011. (SSBSE'11), p. 142–157. ISBN 978-3-642-23715-7. Disponível em: <http://dl.acm.org/citation.cfm?id=2042243.2042264>.

SVAHNBERG, M. et al. A systematic review on strategic release planning models. *Inf. Softw. Technol.*, Butterworth-Heinemann, Newton, MA, USA, v. 52, n. 3, p. 237–248, mar. 2010. ISSN 0950-5849.

TALBI, E.-G. *Metaheuristics: From Design to Implementation*. [S.l.]: Wiley Publishing, 2009. ISBN 0470278587, 9780470278581.

TONELLA, P.; SUSI, A.; PALMA, F. Using Interactive GA for Requirements Prioritization. *2nd International Symposium on Search Based Software Engineering*, Ieee, n. Section II, p. 57–66, sep 2010.

TONELLA, P.; SUSI, A.; PALMA, F. Interactive requirements prioritization using a genetic algorithm. *Information and Software Technology*, Elsevier B.V., v. 55, n. 1, p. 1–15, jul 2013. ISSN 09505849.

VEERAPPA, V.; LETIER, E. Clustering Stakeholders for Requirements Decision- Making. In: Berry, Daniel and Franch, X. (Ed.). *Requirements Engineering: Foundation for Software Quality*. [S.l.]: Springer Berlin Heidelberg, 2011. p. 202–208.

WATERIDGE, J. How can IS/IT projects be measured for success. *International Journal of Project Management*, v. 16, n. 1: 59-63, p. 1998, 1998.

WICKRAMASINGHE W; LI, X. Integrating User Preferences with Particle Swarms for Multi-Objective Optimization. *Proceedings of the conference comp on Genetic and evolutionary computation - GECCO '08*, p. 745–752, 2008.

WIEGERS, K. First Things First: Prioritizing Requirements. *Software Development Online*, v. 7, p. 48–53, set. 1999.

WIERINGA, R. *Design science methodology for information systems and software engineering*. London: Springer Verlag, 2014. [Http://eprints.eemcs.utwente.nl/25449/](http://eprints.eemcs.utwente.nl/25449/).

WOHLIN, C. et al. *Experimentation in Software Engineering*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2012. 236 p. ISBN 9783642290435.

XUAN, J. et al. Solving the large scale next release problem with a backbone-based multilevel algorithm. *IEEE Transactions on Software Engineering*, v. 38, n. 5, p. 1195–1212, Sept 2012. ISSN 0098-5589.

YAVARI, A. Assessment of Effective Risk in Software Projects based on Wallace ' s Classification Using Fuzzy Logic. n. October, p. 58–64, 2013.

ZAVE, P. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 29, n. 4, p. 315–321, dez. 1997. ISSN 0360-0300.

ZHANG, Y.; FINKELSTEIN, A.; HARMAN, M. Search Based Requirements Optimisation : Existing Work & Challenges. In: *Proceeding REFSQ '08 Proceedings of the 14th international conference on Requirements Engineering: Foundation for Software Quality*. [S.l.: s.n.], 2008. p. 88–94.

ZHANG, Y.; HARMAN, M. Search based optimization of requirements interaction management. In: *Proceedings of the 2Nd International Symposium on Search Based Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2010. (SSBSE '10), p. 47–56. ISBN 978-0-7695-4195-2.

ZHANG, Y. et al. Comparing the performance of metaheuristics for the analysis of multi-stakeholder tradeoffs in requirements optimisation. *Information and Software Technology*, Elsevier B.V., v. 53, n. 7, p. 761–773, jul 2011. ISSN 09505849.

ZHANG, Y.; HARMAN, M.; LIM, S. L. Empirical evaluation of search based requirements interaction management. *Information and Software Technology*, Elsevier B.V., v. 55, n. 1, p. 126–152, apr 2013. ISSN 09505849.

ZHANG, Y.; HARMAN, M.; MANSOURI, S. A. The multi-objective next release problem. *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07*, ACM Press, New York, New York, USA, p. 1129, 2007.