

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Marcelo Carlos Benitez dos Santos

**Um modelo de avaliação para projeto de arquitetura de LPS
baseado em busca**

Maringá

2016

Marcelo Carlos Benitez dos Santos

**Um modelo de avaliação para projeto de arquitetura de LPS
baseado em busca**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientadora: Prof^a. Dr^a. Thelma Elita
Colanzi Lopes
Coorientador: Prof. Dr. Edson Alves de
Oliveira Junior

Maringá
2016

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Central - UEM, Maringá, PR, Brasil)

S237m Santos, Marcelo Carlos Benitez dos
Um modelo de avaliação para projeto de arquitetura de LPS Baseado em busca / Marcelo Carlos Benitez dos Santos. -- Maringá, 2016.
[125] f. : il., figs., tabs.

Orientadora: Prof.^a Dr.^a Thelma Elita Colanzi Lopes.
Coorientadora: Prof. Dr. Edson Alves de Oliveira Junior.

Dissertação (mestrado) - Universidade Estadual de Maringá, Centro de Tecnologia, Departamento de Informática, Programa de Pós-Graduação em Ciência da Computação, 2016.

1. Linha de Produto de Software (LPS). 2. Arquitetura de Linha de Produto (ALP). 3. Projeto de Arquitetura de LPS. 4. Arquitetura de Linha de Produto (ALP) - Modelo de Avaliação. I. Lopes, Thelma Elita Colanzi, orient. II. Oliveira Junior, Edson Alves de, coorient. III. Universidade Estadual de Maringá. Centro de Tecnologia. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 23.ed. 004.22

MN-003856

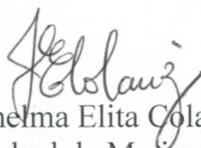
FOLHA DE APROVAÇÃO

MARCELO CARLOS BENITEZ DOS SANTOS

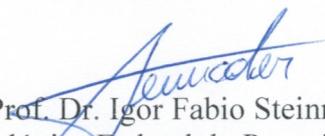
Um modelo de avaliação para projeto de arquitetura de LPS baseado em busca

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

BANCA EXAMINADORA



Profa. Dra. Thelma Elita Colanzi Lopes
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Igor Fabio Steinmacher
Universidade Tecnológica Federal do Paraná – DACOM/UTFPR-CM



Prof. Dr. Igor Scaliante Wiese
Universidade Tecnológica Federal do Paraná – DACOM/UTFPR-CM

Aprovada em: 08 de agosto de 2016.

Local da defesa: Sala 101, Bloco C56, *campus* da Universidade Estadual de Maringá.

AGRADECIMENTOS

Agradeço a Deus por ter abençoado-me nesta caminhada.

Agradeço à minha orientadora, Prof. Dra. Thelma Elita Colanzi Lopes, e a meu coorientador, Prof. Dr. Edson Alves de Oliveira Junior, pela oportunidade de conhecê-los e trabalhar com eles. Em especial, à minha orientadora por sua atenção, dedicação, compreensão e paciência ao longo de toda a excelente orientação recebida.

Agradeço, também, à Prof. Dra. Aline Maria Malachini Miotto Amaral, por ter tido a oportunidade de conhecê-la e por sua dedicação em me ajudar nas orientações.

Aos colegas do DIN, pela solidariedade, preocupação e parceria durante o projeto realizado. Em especial, aos meus amigos Ana, André, Diego, João, Maxmilian, Héber e Heitor, agradeço por dividirem seus conhecimentos e amizades.

À minha família pelo carinho e apoio. Em especial, à minha querida mãe pelo incentivo e por acreditar em mim.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

Um modelo de avaliação para projeto de arquitetura de LPS baseado em busca

RESUMO

A Arquitetura de Linha de Produto (ALP) é considerada o artefato mais importante em uma Linha de Produto de Software (LPS), pois engloba um projeto comum a todos os produtos da LPS. Dessa forma, avaliar e modularizar a ALP é imprescindível para se obter um bom projeto de LPS. No entanto, a avaliação e modularização da ALP são atividades difíceis e que exigem grande esforço humano sem o auxílio de uma ferramenta automatizada. Pode ser fácil para um arquiteto reconhecer um bom projeto de ALP, todavia, tendo em vista as complicações apontadas, torna-se difícil obtê-lo. Diante disso, foi proposta a abordagem denominada MOA4PLA (Multi-Objective Approach for Product-Line Architecture Design), a qual tem o propósito de auxiliar o arquiteto a avaliar e melhorar projetos de ALP. Essa abordagem é multiobjetivo e utiliza um modelo de avaliação composto por funções de avaliação para avaliar o projeto de ALP. Tais funções valem-se de diferentes métricas, as quais envolvem propriedades arquiteturais também diferentes, dentre elas as funções CM(pla) e FM(pla). Atualmente, o modelo de avaliação proposto apresenta problemas em algumas funções de avaliação, pois essas funções avaliam o projeto utilizando métricas com diferentes grandezas, de modo que a informação resultante da avaliação é fornecida de maneira mais genérica. Este trabalho propõe, portanto, reestruturar o modelo de avaliação da MOA4PLA, separando as funções de avaliação CM(pla) e FM(pla) por propriedades arquiteturais, gerando novas funções de avaliação e aumentando as opções de otimização da MOA4PLA. Estudos experimentais foram realizados para analisar uma possível correlação entre as novas funções de avaliação. Os resultados apontam que a maioria das funções são negativamente correlacionadas, de modo que quando uma aumenta, a outra diminui. Sendo assim, a função tem que ser selecionada como objetivo, se o arquiteto quiser priorizá-la no processo de otimização.

Palavras-chave: Linha de Produto de Software (LPS), Projeto de Arquitetura de LPS e Modelo de Avaliação.

ABSTRACT

The Product Line Architecture (ALP) is considered the most important artifact in a Software Product Line (LPS), as it includes a joint design to every LPS product. Thus, evaluate and modularize the ALP is essential to get a good LPS design. However, evaluation and modularization of ALP activities are difficult and require great human effort without the aid of an automated tool. It can be easy for an architect to recognize a good ALP design, however, it is difficult to get it. Therefore, MOA4PLA (Multi-Objective Approach for Product-Line Architecture Design), was proposed, which has the purpose to help the architect to evaluate and improve ALP design. This approach is multi-objective and uses an evaluation model composede by evaluation function to assess the ALP design. These functions are based on different metrics, which involve different architectural properties. Currently, the evaluation model presents problems in some evaluation functions, because these functions evaluate the desingn utilizing metrics with different magnitudes, so that the information resultant of the assessment is provided in a more general way. This work therefore proposes to restructure the evaluation model of MOA4PLA, separating the CM (pla) and FM (pla) by its architectural properties, generating new evaluation functions and increasing optimization options. Experimental studies were performed to investigate a possible correlation between the new evaluation functions. The results show that most of the functions are negatively correlated, so that when one function increases the other decreases. Thus the function has to be selected as an objective, if the architect wants to prioritize it in the optimization process.

Keywords: Software Product Line, Product Line Architecture Design, Evaluation Model.

LISTA DE FIGURAS

2.1	Atividades essenciais de LPS (SEI, 2015)	20
2.2	Dominância de Pareto (Colanzi, 2014)	23
2.3	Processo GA (Sastry et al., 2005)	24
2.4	Processo da MOA4PLA (Colanzi, 2014)	27
2.5	Módulos OPLA-Tool (Colanzi, 2014)	36
3.1	Diagrama de atividades para execução deste trabalho	39
3.2	Projeto do estudo exploratório com NSGA-II	40
3.3	Projeto do estudo exploratório com PAES	41
3.4	Projeto de Estudos de Correlação	43
3.5	Diagramas de atividades para execução deste trabalho	44
4.1	Classes Alteradas na OPLA-Tool	54
5.1	Boxplots dos Resultados de Hipervolume	58
5.2	Experimento 1 - Boxplots DC e EC (AGM)	65
5.3	Experimento 1 - Boxplots DC e EC (BET)	66
5.4	Experimento 1 - Boxplots DC e EC (MOM)	66
5.5	Experimento 2 - Boxplots DC e LCC (AGM)	69
5.6	Experimento 2 - Boxplots DC e LCC (BET)	69
5.7	Experimento 2 - Boxplots DC e LCC (MOM)	70
5.8	Experimento 3 - Boxplots EC e LCC (AGM)	72
5.9	Experimento 3 - Boxplots EC e LCC (BET)	73
5.10	Experimento 3 - Boxplots EC e LCC (MOM)	73
5.11	Experimento 4 - Boxplots LCC e H (AGM)	76
5.12	Experimento 4 - Boxplots LCC e H (BET)	76
5.13	Experimento 4 - Boxplots LCC e H (MOM)	77
5.14	Experimento 5 - Boxplots ACOMP e TAM (AGM)	80
5.15	Experimento 5 - Boxplots ACOMP e TAM (BANK)	80
5.16	Experimento 5 - Boxplots ACOMP e TAM (BET)	81
5.17	Experimento 5 - Boxplots ACOMP e TAM (MOM)	81
5.18	Experimento 6 - Boxplots ACLASS e ACOMP (AGM)	84
5.19	Experimento 6 - Boxplots ACLASS e ACOMP (BET)	84
5.20	Experimento 6 - Boxplots ACLASS e ACOMP (MOM)	85
5.21	Experimento 7 - Boxplots ACOMP e H (AGM)	87
5.22	Experimento 7 - Boxplots ACOMP E H (BET)	88

5.23	Experimento 7 - Boxplots ACOMP e H (MOM)	88
B.1	Projeto de PLA da AGM obtido no Experimento 1	120
B.2	Projeto de PLA da AGM obtido no Experimento 2	121
B.3	Projeto de PLA da AGM obtido no Experimento 3	122
B.4	Projeto de PLA da AGM obtido no Experimento 4	123
B.5	Projeto de PLA da AGM obtido no Experimento 5	124
B.6	Projeto de PLA da AGM obtido no Experimento 6	125
B.7	Projeto de PLA da AGM obtido no Experimento 7	126

LISTA DE TABELAS

Tabela 2.1	Valores Mínimos e Máximos das Métricas	35
Tabela 2.2	Valores Mínimos e Máximos das Funções de Avaliação	36
Tabela 3.1	Informações sobre os projetos de ALP utilizados no estudo	42
Tabela 3.2	Descrição das ALPs Utilizadas no Experimento	45
Tabela 4.1	Funções de Avaliação Propostas	50
Tabela 5.1	Número de Soluções Encontradas	56
Tabela 5.2	Soluções de Menor Distância Euclidiana à Solução Ideal	56
Tabela 5.3	Hipervolume	57
Tabela 5.5	Número Médio de Soluções Encontradas por Experimento	60
Tabela 5.4	Estudos Experimentais das Funções de Avaliação	61
Tabela 5.6	Tabela com <i>Fitness</i> Original Experimentos de Correlação - Parte 1	62
Tabela 5.7	Tabela com <i>Fitness</i> Original Experimentos de Correlação - Parte 2	62
Tabela 5.8	Experimento 1 - Teste de Normalidade - Funções DC x EC	63
Tabela 5.9	Experimento 1 - Teste de Correlação - Funções DC x EC	64
Tabela 5.10	Experimento 2 - Teste de Normalidade - Funções DC x LCC	68
Tabela 5.11	Experimento 02 - Teste de Correlação - Funções DC x LCC	68
Tabela 5.12	Experimento 3 - Teste de Normalidade - Funções EC x LCC	71
Tabela 5.13	Experimento 3 - Teste de Correlação - Funções EC x LCC	72
Tabela 5.14	Experimento 4 - Teste de Normalidade - Métricas H x LCC	75
Tabela 5.15	Experimento 4 - Teste de Correlação - Métricas LCC x H	75
Tabela 5.16	Experimento 5 - Teste de Normalidade - Funções ACOMP x TAM	78
Tabela 5.17	Experimento 5 - Teste de Correlação - Funções ACOMP x TAM .	79
Tabela 5.18	Experimento 6 - Teste de Normalidade - Funções ACLASS x ACOMP	82
Tabela 5.19	Experimento 6 - Teste de Correlação - Funções ACLASS x ACOMP	83
Tabela 5.20	Experimento 7 - Teste de Normalidade - Função ACOMP x H	86
Tabela 5.21	Experimento 7 - Teste de Correlação - Funções ACOMP x H	87
Tabela 5.22	Experimento 1 - Teste de Efeito de Cliff's FM(pla)	90
Tabela 5.23	Experimento 2 - Teste de Efeito de Cliff's FM(pla)	90
Tabela 5.24	Experimento 3 - Teste de Efeito de Cliff's FM(pla)	90
Tabela 5.25	Experimento 4 - Teste de Efeito Cliff's FM(pla)	91
Tabela 5.26	Experimento 5 - Teste de Efeito de Cliff's CM(pla)	91
Tabela 5.27	Experimento 6 - Teste de Efeito de Cliff's CM(pla)	92

Tabela 5.28	Experimento 7 - Teste de Efeito de Cliff's CM(pla)	92
Tabela 5.29	Porcentagem de melhoria NSGA-II (Colanzi et al., 2014)	93
Tabela 5.30	Porcentagem de melhoria Experimento 1 - DCxEC	93
Tabela 5.31	Porcentagem de melhoria Experimento 2 - DCxLCC	94
Tabela 5.32	Porcentagem de melhoria Experimento 3 - ECxLCC	94
Tabela 5.33	Porcentagem de melhoria Experimento 4 - LCCxH	95
Tabela 5.34	Porcentagem de melhoria Experimento 5 - ACOMPxTAM	95
Tabela 5.35	Porcentagem de melhoria Experimento 6 - ACLASSxACOMP	96
Tabela 5.36	Porcentagem de melhoria Experimento 7 - ACOMPxH	96
Tabela 5.37	Modelo de Avaliação da MOA4PLA	98

LISTA DE SIGLAS E ABREVIATURAS

- ACOMP:** Acoplamento de Componentes
- ACLASS:** Acoplamento de Classes
- ALP:** Arquitetura de Linhas de Produto de Software
- AGM:** *Arcade Game Maker*
- CDAC:** *Concern Diffusion over Architectural Components*
- CDAI:** *Concern Diffusion over Architectural Interfaces*
- CDAO:** *Concern Diffusion over Architectural Operations*
- CDepIn:** Dependência de Entrada de uma classe
- CDepOut:** Dependências de Saída de uma Classe
- CIBC:** *Component-level Interlacing Between Concerns*
- CM:** Métricas Convencionais
- DC:** Difusão de Características
- DE:** Distância Euclidiana à Solução Ideal
- DepIn:** Dependências de Entrada de um Pacote
- DepOut:** Dependência de Saída de um Pacote
- DC:** Difusão de Características
- EC:** Entrelaçamento de Características
- FM:** Métricas dirigidas a Características
- GA:** Algoritmos Genéticos
- H:** Coesão Relacional
- HV:** Hypervolume
- IIBC:** *Interface-level Interlacing Between Concerns*
- LCC:** *Lack of Concern-based Cohesion*
- LPS:** Linhas de Produto de Software
- MOA4PLA:** *Multi-Objective Approach for Product-Line Architecture Design*
- MOM:** Mobile Media
- MOEA:** *Multi-objective Evolutionary Algorithms*
- NSGA-II:** *Non-dominated Sorting Genetic Algorithm*
- NumOps:** Número de Operações por interface
- OOBC:** *Operation-level Overlapping Between Concerns*
- OPLA-Tool:** Ferramenta criada para possibilitar a aplicação da abordagem MOA4PLA
- PAES:** *Pareto Archived Evolution Strategy*
- TAM:** Tamanho
- UML:** *Unified Modeling Language*

SUMÁRIO

1	Introdução	15
2	Revisão Bibliográfica	19
2.1	Linha de Produto de Software	19
2.1.1	Arquitetura de Linha de Produto de Software	21
2.2	Engenharia de Software Baseada em Busca	21
2.2.1	Otimização Multiobjetivo	22
2.2.2	Algoritmos Genéticos	23
2.2.3	Indicadores de Qualidade	26
2.3	Otimização de Arquitetura de Linha de Produto de Software	27
2.3.1	Operadores de Busca da MOA4PLA	28
2.3.2	Modelo de avaliação da MOA4PLA	30
2.3.3	Valores das Métricas e Funções	34
2.3.4	Ferramenta OPLA-Tool	36
2.3.5	Considerações Finais	37
3	Materiais e Método de Pesquisa	38
3.1	Sobre o Método de Pesquisa Utilizado Neste Trabalho	38
3.2	Avaliação da Proposta	39
3.2.1	Estudo Exploratório	40
3.2.2	Estudos Experimentais de Correlação	43
3.3	Considerações Finais	48
4	Novo Modelo de Avaliação Proposto para a MOA4PLA	49
4.1	Novas Funções de Avaliação	50
4.2	Implementações das Funções	52
4.3	Considerações Finais	53
5	Estudos Experimentais	55
5.1	Estudo Exploratório	55
5.1.1	Análise e Interpretação dos Resultados	56
5.1.2	Ameaças à validade	57
5.1.3	Discussões	58
5.2	Estudos de Correlação	59
5.2.1	Experimento 1	62

5.2.2	Experimento 2	67
5.2.3	Experimento 3	71
5.2.4	Experimento 4	74
5.2.5	Experimento 5	77
5.2.6	Experimento 6	82
5.2.7	Experimento 7	86
5.2.8	Análise dos Resultados dos Experimentos de Correlação	89
5.2.9	Ameaças à Validade	97
5.3	Considerações Finais	97
6	Conclusão	101
	REFERÊNCIAS	104
A	Artigo publicado no WESB 2015	109
B	Algumas PLAs obtidas nos estudos de correlação	120

Introdução

Linha de Produto de Software (LPS) é uma abordagem com grande destaque no mundo acadêmico e na indústria (SPLC, 2014). Essa abordagem, de acordo com Linden et al. (2007), tem o objetivo de maximizar os lucros e diminuir o tempo de desenvolvimento.

O objetivo da abordagem de LPS é desenvolver uma família de produtos de software para um determinado domínio e reutilizar os artefatos dos produtos de software. Segundo Linden et al. (2007), a abordagem baseia-se no desenvolvimento de um núcleo de artefatos reusáveis, que inclui características obrigatórias e variáveis entre os produtos de software da LPS.

Conforme Kang et al. (1990), uma característica (do inglês *feature*) é uma capacidade do sistema que é relevante e visível para o usuário final. Existem três tipos de características: obrigatórias, variáveis e opcionais. As obrigatórias estão presentes em todos os produtos da LPS; as variáveis (variabilidades) incluem pontos de variação, os quais estão associados às diferentes alternativas de projeto daquela variabilidade; e as opcionais podem ou não fazer parte de um produto.

Essas características devem convergir o máximo possível para serem modulares, visando facilitar a evolução e manutenção da LPS. Modularizar é dividir o software em módulos que realizam uma função completa, independente de outras funções (Pressman, 2006).

Em uma LPS, a Arquitetura de Linha de Produto de Software (ALP) define um projeto comum para todos os produtos a serem derivados da LPS, incluindo as variabilidades entre os produtos. A ALP é considerada o artefato mais importante, pois ela representa a estrutura central do projeto. A ALP contém especificações sobre as características obrigatórias, variáveis e opcionais (Linden et al., 2007).

Conforme Martens et al. (2010), para obter o melhor custo-benefício (*trade-off*) de uma ALP é preciso que seus pacotes estejam, o máximo possível, modulares. No entanto, obter uma ALP modular não é uma tarefa fácil, pois assim como em um projeto de software tradicional, existem várias alternativas para a definição de seus módulos, como exemplo, o acoplamento e a coesão entre eles, as características dos produtos de software distribuídas em mais de um módulo, diversos pontos de variação e variantes, entre outros.

Modularizar uma ALP é um problema que, na maioria das vezes, possui mais de um fator (objetivo) a ser otimizado, e é chamado de multiobjetivo. Nesse caso, é preciso otimizar vários interesses, que, em geral, podem ser conflitantes, na maioria das vezes, quando se melhora um objetivo, piora-se o outro, ocasionando, então, um conjunto de boas soluções. Esses objetivos são definidos através do modelo de avaliação, utilizando um conjunto de métricas arquiteturais para avaliar os atributos da ALP.

As funções de avaliação são desenvolvidas utilizando métricas de software, que são usadas conforme o que se pretende medir, por exemplo: se a intenção é medir o acoplamento entre as classes de um projeto, utilizam-se métricas que medem relacionamentos de entrada e saída de determinada classe.

Encontrar o melhor custo-benefício entre as métricas a serem utilizadas na definição dos objetivos esperados com a otimização do projeto de ALP transforma a obtenção de uma boa ALP em uma tarefa que demanda grande esforço, uma vez que essa tarefa é muito difícil sem a ajuda de uma abordagem automatizada. Para um arquiteto experiente, é fácil reconhecer um bom projeto de ALP, porém é difícil obtê-lo (Colanzi, 2014).

Os autores Aleti et al. (2013), Harman et al. (2009) e Rähkä (2010) apontam que, nesse cenário, a Engenharia de Software Baseada em Busca (*Search Based Software Engineering* (SBSE)) é uma alternativa para solucionar o problema. Técnicas aplicadas na área de SBSE têm mostrado bons resultados em descobrir automaticamente soluções próximas das ideais. Em SBSE, os problemas de Engenharia de Software são formulados como problemas de otimização a serem resolvidos por meio de técnicas baseadas em busca, e têm alcançado resultados positivos (Colanzi, 2014).

Diante disso, Colanzi (2014) propôs uma abordagem de otimização multiobjetivo denominada *Multi-Objective Approach for Product-Line Architecture Design* (MOA4PLA). A MOA4PLA é uma abordagem sistemática e automatizada que usa algoritmos de busca multiobjetivos para avaliar e melhorar o projeto de ALP, em termos de princípios básicos de projeto, modularização de características e extensibilidade de LPS.

A MOA4PLA trabalha produzindo um conjunto com as soluções que apresentam o melhor custo benefício entre os vários objetivos otimizados. Ela possui quatro principais aspectos: (1) engloba um processo para conduzir a otimização do projeto de ALP

por meio de algoritmos multiobjetivos; (2) inclui um metamodelo que representa as ALPs, a fim de viabilizar a manipulação da ALP por parte dos algoritmos; (3) trata a modularização de características por meio de um novo operador de busca; e, (4) propõe um tratamento multiobjetivo para o problema de projeto de ALP, incluindo métricas dirigidas a características (Colanzi, 2014).

Atualmente, o modelo de avaliação da MOA4PLA é composto por quatro funções de avaliação, denominadas CM(pla) (Métricas convencionais), FM(pla) (Métricas dirigidas a características), Ext(pla) (Extensibilidade de ALP) e Eleg(pla) (Elegância), descritos na Seção 2.3.2, que são utilizadas para avaliar cada solução gerada no processo de otimização. Depois de avaliar o conjunto de alternativas de ALPs encontradas, o modelo de avaliação da abordagem apresenta as melhores opções conforme os objetivos definidos pelo arquiteto.

No entanto, duas funções da abordagem MOA4PLA, a CM(pla) e a FM(pla), demonstram um problema, pois elas agregam métricas com diferentes grandezas e fazem uma avaliação mais genérica da ALP (Guizzo, 2014).

Sendo assim, um dos objetivos deste trabalho é refinar o modelo de avaliação original da MOA4PLA, distribuindo as métricas em outras funções, incluindo, dessa forma, a divisão das funções CM(pla) e FM(pla) em novas funções separadas por propriedades arquiteturais. Dessa forma a avaliação fica mais precisa e as funções de avaliação trazem resultados mais condizentes com o propósito de cada métrica. Experimentos computacionais, trazem indícios sobre o comportamento das novas funções de avaliação, por exemplo, se correlacionadas fica mais fácil definir o conjunto de funções que se pretende otimizar.

O outro objetivo é investigar se as novas funções de avaliação são correlacionadas, o que pode direcionar a escolha dos objetivos a serem otimizados. As novas funções são descritas no Capítulo 4. Com essas novas funções de avaliação, o processo de busca será melhor guiado, levando à obtenção de soluções melhor projetadas no que diz respeito aos objetivos definidos pelo usuário. Dessa maneira, o usuário terá mais opções de reestruturação e os resultados tendem a ser mais precisos, podendo utilizar diferentes combinações de funções de avaliação no modelo de avaliação.

Experimentos foram realizados com a intenção de avaliar o desempenho das funções de avaliação propostas neste trabalho. Foram realizados um estudo exploratório e um estudo empírico, composto por sete estudos experimentais. O estudo exploratório foi realizado com a intenção de avaliar o desempenho de dois *Algoritmos de Busca Multiobjetivos* (MOEAs) NSGA-II e PAES com as seis funções de avaliação propostas. No estudo empírico, foram realizados sete estudos experimentais com as funções: ACLASS, ACOMP, TAM, DC, EC e separadamente com as métricas LCC e H pertencentes à função COE,

esses experimentos foram conduzidos para analisar uma possível correlação entre as novas funções de avaliação propostas neste trabalho.

O texto deste trabalho está organizado em capítulos. No Capítulo 2 é apresentada uma revisão da literatura sobre Linhas de Produto de Software, Arquitetura de Linha de Produto de Software, Engenharia de Software Baseada em Busca, Otimização Multiobjetivo, Algoritmos Genéticos, Otimização da Arquitetura de LPS, Funções de Avaliação e Ferramenta OPLA-Tool. O Capítulo 3 descreve o Método de Pesquisa, bem como os Materiais utilizados. O Capítulo 4 trata do Novo Modelo de Avaliação Proposto na Dissertação. O Capítulo 5 detalha os experimentos realizados para analisar as funções de avaliação propostas neste trabalho, assim como os Resultados. No Capítulo 6 são descritas as Conclusões da Dissertação. E, por fim, no Apêndice A encontra-se o artigo publicado no WESB (Software Engineering Based Search) 2015, o qual descreve o Estudo Exploratório, realizado com as seis funções de avaliação aqui propostas.

Revisão Bibliográfica

Este capítulo aborda os principais conceitos envolvidos na discussão apresentada neste trabalho, os quais serão descritos nas próximas seções.

2.1 Linha de Produto de Software

Produtos de Softwares desenvolvidos a partir de uma LPS são baseados em uma família de produtos e possuem o mesmo domínio, segundo Linden et al. (2007). Cada produto da LPS é derivado da ALP, que é comum a todos os produtos da LPS. O intuito da LPS é reduzir custos, oferecer produtos melhores, facilidade na manutenção e diminuir o tempo de desenvolvimento.

O repositório de todos os artefatos do software é chamado de núcleo de artefatos, no qual se encontra a ALP. Por ser o único artefato que abrange toda a LPS, a ALP é considerada o artefato mais importante, de modo que, se a ALP for mal projetada, toda a LPS é afetada. Portanto, é de extrema importância analisar a ALP e avaliar sua qualidade antes das derivações dos produtos (Colanzi, 2014; Oliveira Junior, 2010).

As características da LPS são funcionalidades importantes, visíveis para o usuário final, pois são essas características que diferenciam um produto do outro, conforme apontam Kang et al. (1990).

Segundo Linden et al. (2007), existem características:

- **Obrigatórias:** presentes em todos os produtos da LPS;
- **Opcionais:** que podem ou não estar presentes em um produto da LPS;

- **Alternativas:** que formam um conjunto de características em que uma delas estará no produto da LPS.

As variações são os pontos mais importantes em uma LPS, as quais diferenciam uma arquitetura padrão de uma ALP, pois fazem com que o arquiteto considere todos os produtos que podem ser derivados da ALP. Essas variações são chamadas de variabilidades. Os artefatos que preenchem os pontos de variação são as variantes, isto é, as alternativas de projeto para resolver o ponto de variação (Linden et al., 2007).

O desenvolvimento da LPS possui três atividades essenciais: desenvolvimento do núcleo de artefatos, desenvolvimento do produto e o gerenciamento deste. O desenvolvimento do núcleo de artefatos corresponde à engenharia de domínio, na qual são realizados a análise do produto, o projeto de domínio e a implementação do domínio. O desenvolvimento do produto corresponde à engenharia de aplicação, em que são feitas etapas de análise dos requisitos, a configuração do produto, a customização e o desenvolvimento do produto. Por fim, o gerenciamento da LPS garante o processo e a evolução dos produtos da LPS.



Figura 2.1: Atividades essenciais de LPS (SEI, 2015)

A Figura 2.1 ilustra as três atividades essenciais da LPS, que são extremamente relacionadas, de modo que a alteração em qualquer uma delas impacta diretamente nas demais, mostrando um alto *feedback*, mantendo o núcleo de artefatos e o núcleo de desenvolvimento sempre atualizados, segundo Linden et al. (2007). Uma vez que este trabalho está focado na otimização do projeto de ALP, o desenvolvimento do núcleo de artefatos é seu foco principal.

2.1.1 Arquitetura de Linha de Produto de Software

Desenvolver um software que atenda exatamente aos seus propósitos depende de uma boa orientação sobre como desenvolvê-lo, sendo assim, uma arquitetura bem projetada é fundamental para o sucesso do projeto (Schmid e Verlage, 2002).

A ALP é o resultado da engenharia de domínio, que descreve uma arquitetura genérica da LPS e fornece uma solução para os produtos da LPS. Dessa maneira, a ALP representa a abstração de todas as possíveis arquiteturas de produtos que a LPS pode gerar, permitindo que os desenvolvedores reusem pacotes nos diferentes produtos da LPS. Sendo assim, a ALP define as partes essenciais da infraestrutura de reuso e garante que, tanto os pacotes compartilhados, como os específicos para alguns produtos, se encaixem para criar cada produto da LPS. São as variabilidades que permitem aumentar o reuso na LPS. Conforme Schmid e Verlage (2002), quanto mais modularizadas estiverem as variabilidades, maior a capacidade de reuso, derivação de produtos e evolução da LPS.

A avaliação da ALP deve ser considerada uma atividade fundamental durante o ciclo de vida da LPS, pois uma vez bem planejada, trará todos os benefícios citados anteriormente (Taylor et al., 2010). Segundo Colanzi (2014), o projeto de ALP é uma atividade complexa, para a qual há mais de uma possível solução boa ou aceitável. Devido à complexidade do problema, a SBSE procura trazer soluções a partir de algoritmos multiobjetivos, o que será explorado na próxima seção.

2.2 Engenharia de Software Baseada em Busca

A Engenharia de Software Baseada em Busca (SBSE) (Harman et al., 2009) é um campo de pesquisa da engenharia de software que utiliza técnicas baseadas em busca para resolver diferentes problemas da Engenharia de Software. SBSE contribui para reduzir esforços e custos associados ao desenvolvimento de software desde que as soluções encontradas pela busca satisfaçam as restrições técnicas que estão geralmente em conflito e, em geral, são muito difíceis de ser obtidos por engenheiros de software (Vergilio et al., 2011).

De acordo com Harman et al. (2009), em SBSE os problemas da Engenharia de Software são tratados como problemas de otimização, em que o objetivo é minimizar ou maximizar uma função ou grupo de fatores. Dessa forma, para os autores, as técnicas de otimização baseadas em busca geralmente estão associadas a dois aspectos: um espaço de busca, com todas as possíveis soluções para o problema; e uma função de avaliação, que avalia a qualidade da solução.

Como mencionado anteriormente, o problema de projeto de ALP pode ser considerado um problema multiobjetivo. Por isso, na próxima seção descrevem-se os conceitos básicos de otimização multiobjetivo. E na Seção 2.2.2, serão abordados os algoritmos genéticos, os quais já foram adaptados para tratar problemas multiobjetivos. Na Seção 2.2.3, por sua vez, são apresentados em indicadores de qualidade utilizados nos estudos experimentais.

2.2.1 Otimização Multiobjetivo

Problemas multiobjetivos envolvem mais de um objetivo e conseqüentemente mais de uma função de avaliação, sendo que estes estão diretamente ligados a objetivos conflitantes. Portanto, a ideia de encontrar uma única solução boa geralmente não pode ser aplicada, pois mais de uma solução boa ou aceitável podem existir. Algoritmos multiobjetivos permitem utilizar mais de uma função de avaliação para resolver problemas multiobjetivos. Esses problemas necessitam de otimização simultânea, pois tratam de diferentes interesses, conseqüentemente, não há uma única solução. Nesse sentido, segundo Coello et al. (2007), encontrar um conjunto de soluções que representem os diversos interesses e compromissos entre eles é a melhor alternativa.

Para encontrar o conjunto de soluções candidatas a solucionar os problemas multiobjetivos, pode-se utilizar o conceito de dominância de Pareto (Pareto, 1927). Este se aplica às soluções dominadas e não dominadas, sendo elas:

- **Dominadas:** Uma solução A domina uma solução B quando todos os valores de objetivos de A forem menores ou iguais aos valores de B e existe pelo menos um valor de objetivo de A que seja menor que seu correspondente valor em B;
- **Não dominadas:** Uma solução A domina uma solução B quando todos os valores de objetivos de A forem maiores ou iguais aos valores de B e existe pelo menos um valor de objetivo de B que seja maior que seu correspondente valor em A.

A Figura 2.2 ilustra como minimizar dois fatores (objetivos) para escolher um meio de transporte, levando em consideração o tempo e o valor desse transporte. Nelas estão contidas as opções como: avião, carro, trem, ônibus e moto, representados respectivamente pelas letras “a, c, d, f e g”, que formam a PFapprox. As opções b (táxi) e e (navio) são dominadas por outras opções melhores, portanto devem ser descartadas. As opções táxi e navio são dominadas por outras soluções com valores menores como avião e trem, pois em pelo menos um dos objetivos existem opções melhores. Nesse caso, táxi é dominado pelo carro, pois, apesar de possuírem o mesmo objetivo tempo, no objetivo valor, a opção carro é menor (Colanzi, 2014).

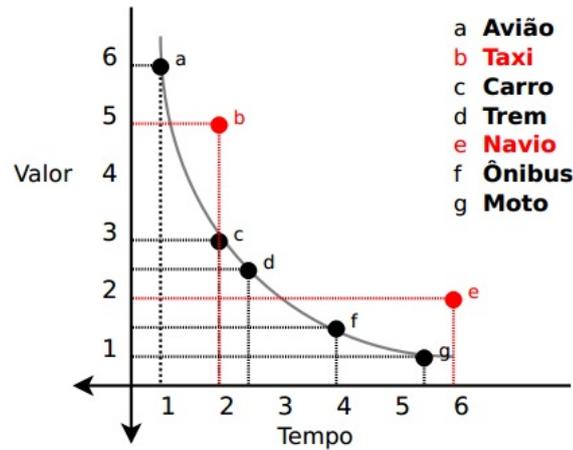


Figura 2.2: Dominância de Pareto (Colanzi, 2014)

A Otimização Multiobjetivo consiste em otimizar simultaneamente vários interesses interdependentes e muitas vezes conflitantes e, por isso, não há uma única solução. Assim, o objetivo é encontrar o conjunto de soluções que melhor representa os diversos interesses definidos nos objetivos. Para isso é utilizado o conceito de dominância de Pareto (Pareto, 1927) que permite comparar soluções considerando todos os objetivos do problema. As soluções são chamadas de dominadas e não-dominadas. As não dominadas são as melhores e formam uma aproximação à fronteira de Pareto (PF_{approx}). A união de todos os PF_{approx} , excluindo as soluções dominadas, formam o PF_{known} , que representa as melhores soluções encontradas por um algoritmo para um determinado problema.

2.2.2 Algoritmos Genéticos

Algoritmos genéticos (GA) são algoritmos baseados na teoria da evolução, segundo a qual os melhores indivíduos se sobressaem em relação aos demais. Esses indivíduos sobrevivem e se reproduzem mais, de modo que nessa reprodução podem ocorrer mutações, e isso implica que indivíduos melhores ou piores podem surgir (Coello et al., 2007).

A Figura 2.3 ilustra o processo básico do funcionamento de um GA, o qual pode ser descrito como (Sastry et al., 2005):

1. **População Inicial:** A população inicial com soluções candidatas costumam ser geradas aleatoriamente, dentro do espaço de estados, apesar de o conhecimento específico do domínio poder ser facilmente incorporado;

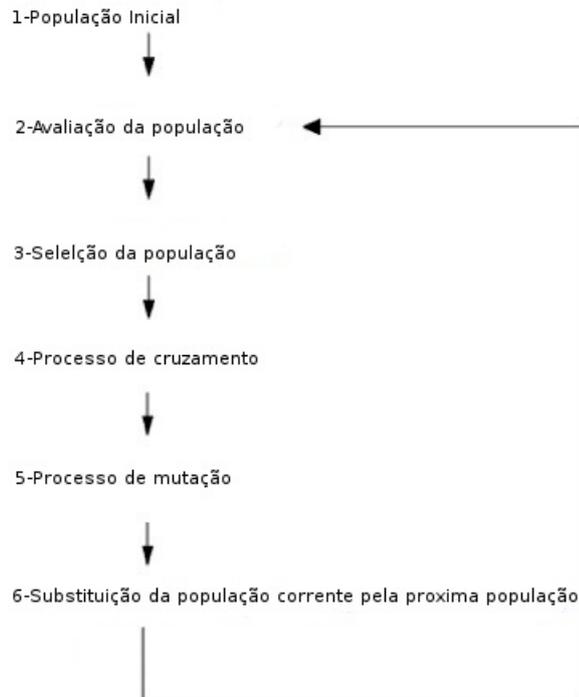


Figura 2.3: Processo GA (Sastry et al., 2005)

2. **Avaliação da População:** Uma vez que a população é inicializada ou que uma nova população descendente é criada, avalia-se o valor de aptidão das soluções candidatas;
3. **Seleção da População:** A seleção aloca probabilisticamente mais cópias daquelas soluções com os melhores valores de aptidão, dessa forma, as melhores soluções candidatas sobreviverão na próxima população. A ideia principal da seleção é preferir as melhores soluções em detrimento das piores, como ocorre na seleção genética humana proposta por Darwin (2004), em que os indivíduos mais aptos sobrevivem. Os principais métodos de seleção se valem dessa ideia, como a seleção por roleta, por torneio e por ranking;
4. **Cruzamento:** O cruzamento combina partes de duas soluções (pais) para criar uma nova solução (filho), possivelmente com melhor aptidão. Há várias formas de fazer o cruzamento, e o mecanismo projetado pode ser específico para o tipo de representação das soluções candidatas;
5. **Mutação:** Enquanto o cruzamento atua sobre dois ou mais cromossomos, a mutação modifica aleatoriamente a solução criada pelo cruzamento (filho). Há várias formas de se realizar a mutação, mas geralmente ela executa uma varredura aleatória na vizinhança da solução candidata;

- 6. Substituição da População Corrente pela Próxima População:** A população descendente criada a partir da seleção, cruzamento e mutação substitui a população corrente. Dentre as técnicas de substituição pode-se citar, por exemplo, elitismo, em que somente as soluções com as melhores aptidões permanecem na população descendente;
7. Repetir os passos 2-6 até que a condição de parada seja alcançada. A condição de parada pode ser dada, por exemplo, por um número máximo de gerações ou por número de avaliações de aptidão.

Como pode ser observado no processo, GAs utilizam operadores de busca para movimentar os indivíduos. Os movimentos básicos são realizados pelos operadores de cruzamento de indivíduos e pelos operadores de mutação dos novos indivíduos gerados. Dessa maneira, considerando as taxas de probabilidade de ocorrer o cruzamento e/ou a mutação, a população inicial apresenta evolução geração a geração, por meio da aplicação dos operadores de cruzamento e mutação (Burke e Kendall, 2005).

Alguns algoritmos genéticos foram adaptados para resolver problemas multiobjetivos. Esses algoritmos são chamados de *Multi-Objective Evolutionary Algorithms* (MOEAs). Dentre eles o mais conhecido é o NSGA-II. Neste trabalho serão utilizados dois MOEAs o NSGA-II e o PAES, pois são algoritmos utilizados para otimização multiobjetivo.

Segundo Deb et al. (2002), *Non-dominated Sorting Genetic Algorithm* (NSGA-II) é um MOEA baseado em GA com forte estratégia de elitismo. No NSGA-II as opções de reestruturação são chamadas de população e são ordenadas em fronteiras de soluções não dominadas. Existem diversas fronteiras que são geradas, e ele trabalha ordenando os indivíduos da população de pais e filhos de acordo com a dominância entre as soluções. As soluções não dominadas formam a primeira fronteira, ao passo que a segunda fronteira é formada pelas soluções que se tornaram não dominadas (filhos). Da mesma forma, a terceira fronteira é o resultado das não dominadas da primeira e da segunda, e assim por diante, até que todas as soluções permaneçam na fronteira.

De acordo com Knowles e Corne (2000), no algoritmo *Pareto Archived Evolution Strategy* (PAES) sobra apenas uma solução em cada geração, e os indivíduos novos são obtidos somente através dos operadores de mutação. Esse conceito de população é diferente das demais estratégias tradicionais dos GAs. Em cada geração o PAES cria uma nova solução filho, que é comparada com a solução pai. Se a solução filho é dominada pela solução pai, a solução filho é descartada. No entanto se a solução filho domina a solução pai, aquela toma o lugar desta. Porém, um arquivo externo de soluções é populado com as soluções não dominadas encontradas durante o processo.

2.2.3 Indicadores de Qualidade

Os indicadores de qualidade são utilizados para avaliar o desempenho dos algoritmos de busca multiobjetivos, podendo comparar os resultados obtidos por diferentes algoritmos de busca multiobjetivos. Esses indicadores, geralmente, baseiam-se em três conjuntos de soluções encontrados por estes algoritmos (Colanzi, 2014):

- PF_{approx} : Formado pelas soluções não dominadas de uma execução de um algoritmo. Os algoritmos de busca apresentam uma característica não determinística, dessa maneira, são executadas várias rodadas para o mesmo problema. Nesse sentido, se forem executadas 30 rodadas, são obtidos 30 conjuntos de PF_{approx} ;
- PF_{known} : Formado pela agregação de todos os PF_{approx} obtidos por um algoritmo, com exceção das soluções dominadas e repetidas. Desse modo, PF_{known} corresponde às melhores soluções encontradas por um algoritmo para um dado problema;
- PF_{true} : Formado pela fronteira de Pareto ótima para um problema. Porém, em situações nas quais a fronteira não é conhecida, é recomendado que o conjunto PF_{true} seja a agregação de todas as soluções encontradas em todas as rodadas de todos os algoritmos que tiverem sido executados, com exceção das soluções dominadas e repetidas (Zitzler et al., 2003).

Os de indicadores de qualidade que foram utilizados neste trabalho são Hipervolume (HV) (Zitzler et al., 2003) e Distância Euclidiana à Solução Ideal (ED) (Cochrane e Zeleny, 1973). O Hipervolume (Zitzler et al., 2003) é utilizado para medir resultados quantitativamente e não dominados de um problema multiobjetivo, ou seja, este indicador mede as fronteiras de Pareto obtidas como resultado. Com isso, o indicador compara dois conjuntos PF_{approx} , de modo que se um PF_{approx} domina outro PF_{approx} o resultado é o HV do primeiro conjunto maior que o segundo conjunto. Se o problema for de minimização, quanto maior for o valor de HV, melhor é a fronteira de Pareto, pois HV domina a maior parte do espaço de busca.

A Distância Euclidiana é calculada para apoiar a tomada de decisões, tem a finalidade de encontrar a solução mais próxima da ideal. Uma solução é considerada ideal quando apresenta menor valor para cada objetivo, considerando um problema de minimização (Cochrane e Zeleny, 1973). As soluções que formam o PF_{true} são as soluções consideradas ideais.

2.3 Otimização de Arquitetura de Linha de Produto de Software

Segundo Colanzi (2014), MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*) é uma abordagem que tem a finalidade de automatizar a busca por melhores projetos de ALP. Ela utiliza algoritmos evolutivos multiobjetivos para avaliar e melhorar projetos de ALP. Ainda segundo a autora, essa abordagem traz um conjunto com as soluções que possuem o melhor custo-benefício entre os vários objetivos otimizados, em termos de princípios básicos de projeto, modularização de características e extensibilidade de LPS. A abordagem é constituída de quatro atividades que são apresentadas na Figura 2.4 a seguir:

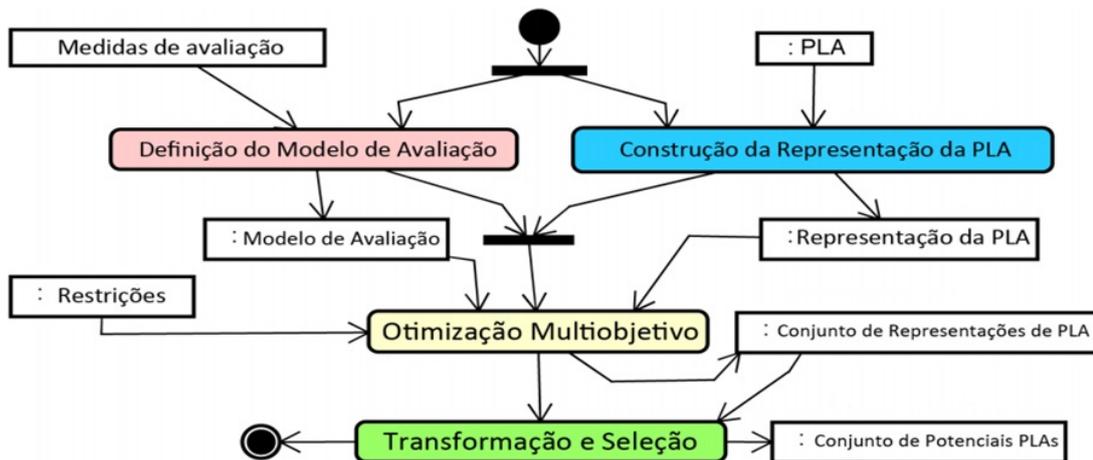


Figura 2.4: Processo da MOA4PLA (Colanzi, 2014)

1. **Construção da Representação:** Esta atividade recebe como entrada um projeto de ALP modelado como um diagrama de classes UML (*Unified Modeling Language*) estereotipado. O objetivo dessa atividade é instanciar uma solução inicial, a partir do projeto de ALP, para ser utilizada pelo algoritmo como ponto de partida do processo evolutivo. A saída dessa atividade é uma representação da ALP, contendo todos os elementos arquiteturais. A ALP pode ser utilizada contendo qualquer estereotipo. Segundo Harman et al. (2009), a atividade Construção da Representação é o primeiro ingrediente chave necessário para atacar um problema usando algoritmos de busca;
2. **Definição do Modelo de Avaliação:** Nesta atividade, o usuário define quais medidas devem ser utilizadas no modelo de avaliação para a otimização do projeto

ALP. Dessa maneira, o usuário define as métricas que atendem às necessidades do projeto, métricas essas que constituem os objetivos a serem otimizados. Utilizar diferentes tipos de métricas ajuda o usuário na análise do melhor custo-benefício entre os diferentes atributos de qualidade, pois algumas delas podem ser conflitantes. A MOA4PLA inclui quatro funções de avaliação compostas por várias métricas. A primeira é a CM(pla), cujo objetivo é contribuir para alcançar soluções propensas a conter projetos adequados, do ponto de vista de alta coesão e baixo acoplamento. A segunda é a FM(pla), que tem o objetivo de permitir que projetos com alta modularização de características sejam alcançados pela MOA4PLA. A terceira é a Ext(pla), que objetiva maximizar a extensibilidade da ALP. A última função é a Eleg(pla), que tem o objetivo de melhorar a elegância do projeto de ALP alcançado em cada solução gerada pela MOA4PLA. Segundo Harman et al. (2009), a atividade Definição do Modelo de Avaliação é o segundo ingrediente chave necessário para atacar um problema usando algoritmos de busca. O modelo de avaliação da MOA4PLA é descrito com detalhes na Seção 2.3.2;

3. **Otimização Multiobjetivo:** Nesta atividade o projeto recebido como entrada é otimizado no processo de busca de acordo com os operadores de busca definidos pela abordagem MOA4PLA. Esses operadores de busca realizam movimentos básicos dos algoritmos evolutivos e são eles os operadores de cruzamento de indivíduos e operadores de mutação dos novos indivíduos gerados. Cada opção de projeto obtida é avaliada conforme o modelo de avaliação, sendo assim, gerado um conjunto com as opções de projeto de ALP com o melhor custo-benefício como saída. Os operadores de mutação e cruzamento da MOA4PLA são descritos com detalhes na Seção 2.3.1;
4. **Transformação e Seleção:** Esta atividade recebe como entrada o conjunto de soluções geradas como saída da otimização multiobjetivo. Essas soluções são as alternativas de projeto encontradas pela MOA4PLA e são convertidas em uma forma legível para o usuário.

2.3.1 Operadores de Busca da MOA4PLA

Esta seção tem por objetivo apresentar operadores de busca utilizados na representação adotada pela MOA4PLA. A abordagem possui cinco operadores de busca convencionais, que são descritos a seguir (Colanzi, 2014):

1. **Move Method:** Dadas duas classes selecionadas aleatoriamente, esse método move um método selecionado aleatoriamente entre essas classes. Além de remover

o método da classe original e inseri-lo na classe destino, o operador adiciona uma associação bidirecional entre essas duas classes. O operador só é aplicado satisfazendo algumas condições, em relação à classe de origem: ela não deve fazer parte de uma hierarquia de herança; ela deve possuir mais de um atributo e mais de um método; um método não deve estar associado com a característica da variabilidade vinculada à classe, caso ela seja um ponto de variação ou uma variante;

2. **Move Attribute:** Dadas duas classes selecionadas aleatoriamente, esse método move um atributo selecionado aleatoriamente entre duas classes. Ele também adiciona uma associação bidirecional entre essas duas classes. O operador só é aplicado satisfazendo algumas condições em relação a classe de origem: ela não deve fazer parte de uma hierarquia de herança, se tiver mais de um atributo e mais de um método e, se o atributo não estiver associado com a característica da variabilidade vinculada à classe, caso ela seja um ponto de variação ou uma variante;
3. **Add Class:** Dada uma determinada classe de um pacote, esse operador move um atributo ou um método selecionado aleatoriamente para uma nova classe, criado em um pacote selecionado aleatoriamente. Além disso, caso as classes sejam do mesmo pacote, o operador adiciona uma associação bidirecional entre essas duas classes. Uma interface do pacote destino é adicionada como interface requerida para o pacote de origem da classe, caso os pacotes são diferentes. Esse operador é aplicado satisfazendo algumas condições: a classe não deve fazer parte de uma hierarquia de herança; a classe deve possuir mais de um atributo e mais de um método; os pacotes das interfaces devem ser da mesma camada arquitetural;
4. **Move Operation:** Dadas duas interfaces selecionadas aleatoriamente, o operador (Move Operation) move uma operação entre elas. O operador adiciona a interface destino como interface realizada pelo pacote da interface original da operação movida, caso os pacotes que realizam as interfaces forem diferentes. A operação é movida se os pacotes das interfaces forem da mesma camada arquitetural;
5. **Add Component:** Dado um operador selecionado aleatoriamente de uma interface de um pacote, esse operador move essa operação para uma nova interface de um novo pacote que é criado na arquitetura. Além de tirar a operação da interface original e inseri-la na nova interface, o operador adiciona a nova interface como interface realizada pelo pacote da interface de origem. O novo pacote é criado na mesma camada, com o mesmo sufixo do pacote de origem.

A abordagem tem também um operador de busca dirigido à características, que é descrito a seguir:

1. **Feature-driven Operator:** Dado um pacote selecionado aleatoriamente, o operador objetiva modularizar uma característica que esteja entrelaçada com outra(s) nesse pacote. Partindo do pressuposto de que uma característica da LPS costuma ser resolvida por um grupo de elementos arquiteturais, esse método seleciona aleatoriamente um pacote, e, se esse pacote tiver interface(s) associada(s) à diferentes características, ele seleciona uma dessas características para ser modularizada em um novo pacote.

Por meio de operadores de mutação e cruzamento que foram propostos especificamente para a MOA4PLA (Colanzi, 2014), novos indivíduos são gerados utilizando algoritmos genéticos.

O operador de mutação faz mudanças aleatórias nas características dos cromossomos, com a intenção de obter maior diversidade genética na população. No cruzamento, dois cromossomos “pais” são combinados para formarem novos cromossomos “filhos”. Os pais são selecionados aleatoriamente a partir dos cromossomos existentes na população, os pais geralmente são os que tiveram melhor aptidão. Dessa maneira, espera-se que os filhos gerados herdem genes bons, com maior aptidão. Através da aplicação do operador de cruzamento são esperados uma população com cromossomos com melhores *fitness* (Harman et al., 2009). Os algoritmos evolutivos utilizam funções de avaliação (modelo de avaliação) para avaliar qual a melhor solução com base nos objetivos definidos pelo arquiteto. Dessa maneira a MOA4PLA parte de um projeto inicial e melhora o projeto utilizando os algoritmos evolutivos.

2.3.2 Modelo de avaliação da MOA4PLA

Métricas podem ser utilizadas para avaliar arquiteturas de LPS com o intuito de estimar certas propriedades do projeto.

A definição do modelo de avaliação abrange a seleção de métricas e a construção das funções de avaliação. Modelos de avaliação em outros tópicos de SBSE têm sido amplamente estudados, por exemplo: os modelos propostos para o teste de software. No entanto, a definição de um modelo de avaliação para projeto de ALP baseado em busca é uma questão de pesquisa explorada inicialmente pela MOA4PLA. Isso porque a proposta de métricas para avaliar ALPs na engenharia de LPS ainda é uma área de pesquisa recente e o projeto de ALP baseado em busca é um campo de pesquisa incipiente.

O principal objetivo da MOA4PLA é alcançar um projeto de ALP extensível, modular e altamente reutilizável. Depois de investigar as métricas que englobam essas metas, Colanzi (2014) propôs um conjunto mínimo de métricas para estar disponível em sua abordagem, a fim de facilitar a definição do modelo de avaliação por arquitetos. Foi então proposto o primeiro modelo de avaliação para projeto de ALP, baseado em busca no contexto da MOA4PLA, que inclui quatro funções de avaliação: FM(pla), CM(pla), Ext(pla) e Eleg(pla), as quais serão descritas a seguir.

Função CM(pla)

As métricas para avaliar princípios básicos de projeto, tais como baixo acoplamento e alta coesão, são denominadas como convencionais. A função de avaliação CM(pla), apresentada na Equação 2.1 é composta pela soma de várias métricas convencionais. Nela **c** representa o número de pacotes, **itf** representa o número de interfaces e **cl** representa o número de classes de uma dada ALP.

$$CM(alp) = \sum_{i=1}^c DepIn + \sum_{i=1}^c DepOut + \sum_{i=1}^{cl} CDepIn + \sum_{i=1}^{cl} CDepOut + \frac{\sum_{i=1}^c DepPack}{c} + \frac{\sum_{i=1}^{itf} NumOps}{itf} + \frac{1}{\sum_{i=1}^c H} \quad (2.1)$$

De acordo com Colanzi (2014), as métricas utilizadas na função CM(pla) são:

1. **DepIn** (Dependências de Entrada) (Wüst, 2014): Mede o número de dependências UML onde o pacote é o fornecedor;
2. **DepOut** (Dependências de Saída) (Wüst, 2014): Mede o número de dependências UML onde o pacote é o cliente;
3. **CDepIn** (Dependências de Entrada de uma Classe) (Wüst, 2014): Representa as dependências de entrada de uma classe. Conta o número de elementos arquiteturais que dependem dessa classe;
4. **CDepOut** (Dependências de Saída de uma Classe) (Wüst, 2014): Representa as dependências de Saída de uma Classe. Conta o número de elementos arquiteturais dos quais essa classe depende;
5. **DepPack** (Dependência de Pacotes) (Wüst, 2014): Mede a dependência de pacotes entre classes e interfaces. Um pacote P depende de um pacote P' se neste pacote

uma classe ou interface em P depende de uma classe ou interface em P' ou se existir uma dependência na UML entre P e P' ;

6. **NumOps** (Número de Operações por Interface) (Wüst, 2014): Mede o número de operação por interface. Essa métrica mostra aspectos de reusabilidade de uma ALP porque interfaces pequenas são, em geral, mais facilmente reutilizadas;
7. **H** (Coesão Relacional) (Wüst, 2014): A métrica H mede a coesão relacional do projeto contando o número médio de relacionamentos entre classes e interfaces dentro de um pacote. Considerando N como sendo o número de classes e interfaces dentro do pacote e R como sendo o número de relacionamentos internos ao pacote, temos que: $H = (R + 1) / N$, onde o valor um é somado a R para evitar que H seja 0 quando N é 1. A métrica H auxilia os projetistas a identificar pacotes que possuem elementos pouco relacionados, o que pode significar que os pacotes são pouco coesos.

Na função $CM(pla)$ são utilizadas a média de DepPack de todos os pacotes de ALP e a média de NumOps de todas as interfaces do projeto.

Função $FM(pla)$

A função de avaliação $FM(pla)$ é uma agregação das várias métricas dirigidas a características. $FM(pla)$ consiste da soma dos somatórios de cada métrica dirigida às características. Essa função é usada para avaliar o grau de modularização de uma ALP em termos de características. Na função de avaliação $FM(pla)$, apresentada na Equação 2.2 **i** representa o número de interfaces e **f** representa o número de características de uma dada ALP Colanzi (2014).

$$FM(alp) = \sum_{i=1}^c LCC + \sum_{i=1}^f CDAC + \sum_{i=1}^f CDAI + \sum_{i=1}^f CDAO + \sum_{i=1}^f CIBC + \sum_{i=1}^f IIBC + \sum_{i=1}^f OOBC \quad (2.2)$$

As métricas utilizadas na função $FM(pla)$ são:

1. **LCC** (Lack of Concern-based Cohesion) (Sant'Anna, 2008): Conta o número de interesses (requisitos, funcionais ou não de um sistema de software) com os quais um dado pacote está associado, somado ao número de interesses distintos com os quais as interfaces do pacotes estão associadas, somado ao número de interesses distintos com os quais as operações dessas interfaces estão associadas;

2. **CDAC** (Concern Diffusion over Architectural Components) (Sant'Anna, 2008): Utilizada para medir a modularização de interesses em nível de pacotes;
3. **CDAI** (Concern Diffusion over Architectural Interfaces) (Sant'Anna, 2008): Conta o número de interfaces que contribuem para a realização de um dado interesse. Ela leva em consideração as interfaces que contribuem inteiramente para o interesse e aquelas que possuem pelo menos uma operação associada ao interesse;
4. **CDAO** (Concern Diffusion over Architectural Operations) (Sant'Anna, 2008): Conta o número de operações que contribuem para a realização de um dado interesse;
5. **CIBC** (Component-level Interlacing Between Concerns) (Sant'Anna, 2008): Conta o número de interesses com os quais um dado interesse compartilha ao menos um pacote;
6. **IIBC** (Interface-level Interlacing Between Concerns) (Sant'Anna, 2008): Conta o número de interesses com os quais um dado interesse compartilha ao menos uma interface;
7. **OOBC** (Operation-level Overlapping Between Concerns) (Sant'Anna, 2008): Conta o número de interesses com os quais um dado interesse compartilha ao menos uma operação.

Como as métricas dividem a mesma abordagem de interesse entende-se que a diminuição de uma pode diminuir a outra. Segundo Colanzi (2014), considerando cada característica da LPS como um interesse, as métricas sensíveis a interesses podem ser utilizadas para fornecer indicadores a respeito da modularização de características. Assim, essas métricas (LCC, CDAC, CDAI, CDAO, CIBC, IIBC e OOBC) permitem medir coesão e acoplamento baseados em suas características. Dado que as referidas métricas são aplicadas na MOA4PLA com o intuito de medir a modularização de características, elas serão chamadas, daqui em diante, de métricas dirigidas a características.

Função Ext(pla)

Oliveira Junior et al. (2013) definiu **Extens(pla)** com a finalidade de somar a capacidade de extensão de cada um dos pacotes da ALP. A Equação 2.3 representa o valor invertido da métrica Extens(pla). Esse objetiva maximizar a extensibilidade da ALP, sabendo que os objetivos são minimizados na abordagem MOA4PLA (Colanzi, 2014).

$$Ext(alp) = \frac{1}{ExtensPLA} \quad (2.3)$$

Função ELEG(pla)

O objetivo da função ELEG(pla) é melhorar a elegância do projeto de ALP alcançado em cada solução gerada pela MOA4PLA. Para isso, devem-se minimizar os valores das métricas NAC, EC e ATMR. A Equação 2.4 apresenta função de avaliação proposta por Colanzi (2014):

$$Eleg(alp) = NAC(pla) + EC(pla) + ATMR(pla) \quad (2.4)$$

1. **NAC** (Elegância de números entre classes) (Simons e Parmee, 2012): Mede a elegância de números entre classes. Desvio padrão dos números de atributos e métodos entre as classes de um projeto;
2. **EC** (Elegância de acoplamentos externos) (Simons e Parmee, 2012): Mede a elegância de acoplamentos externos das classes de um projeto;
3. **ATMR** (Elegância da razão entre atributos e métodos) (Simons e Parmee, 2012): Desvio padrão da razão entre atributos e métodos dentro das classes de um projeto.

Essa função é uma agregação dessas métricas de elegância.

2.3.3 Valores das Métricas e Funções

Nesta seção são apresentados os valores mínimos e máximos das métricas e das funções de avaliação utilizadas na abordagem MOA4PLA obtidas através de um experimento realizado por Colanzi (2014). A Tabela 2.1 apresenta os valores mínimos e máximos de cada métrica extraídos durante a execução de um experimento. Foram utilizadas 9 ALPs (AGM-1, AGM-2, AGM-3, AGM-4, LPS-BET, MOM-1, MOM-2, MOM-3 e MOM-4).

Por exemplo, as métricas destacadas em negrito CDAC, CDAI, CDAO, CIBC, IIBC, OOBC e LCC são métricas pertencentes à função de avaliação FM(pla), enquanto a métrica OOBC tem o valor máximo de 6.0, a métrica CDAC tem o valor mínimo de 16.0. A métrica IIBC tem o valor mínimo de 2.0 e o máximo de 59.0. A métrica CDAO tem em seu valor mínimo 80.0 e o máximo de 908.0. Ou seja, são valores que iniciam de dezenas, como o valor mínimo para IIBC (2.0) e vão até centenas, como é o caso do valor

máximo de CDAO (908.0). Com isso fica evidente que estas métricas possuem grandezas diferentes e estão se misturando. Conseqüentemente, uma métrica exerce influência maior que a outra na função de avaliação. Por exemplo, a CDAO que possui valores maiores exerce maior influência na medição de FM(pla) em relação à IIBC.

A diferença nos valores das métricas é menor para a função de avaliação CM(pla). As métricas também destacadas em negrito: H, DepPack, NumOps, DepIn, DepOut, CDepIn e CDepOut, mostram que a métrica H tem valores na casa dos decimais, como valor mínimo de 0.062 e valor máximo de 0.33. A métrica com os valores maiores é a CDepIn, com valor mínimo de 8.0 e máximo de 20.0.

Tabela 2.1: Valores Mínimos e Máximos das Métricas

Métrica	Valor Mínimo	Valor Máximo	Métrica	Valor Mínimo	Valor Máximo
CDAC	16.0	178.0	DepPack	0.0	0.0
CDAI	20.0	170.0	NumOps	4.0	6.666
CDAO	80.0	908.0	DepIn	0.0	0.0
CIBC	12.0	121.0	DepOut	0.0	0.0
IIBC	2.0	59.0	CDepin	8.0	20.0
OOBC	0.0	6.0	CDepOut	1.0	7.0
LCC	11.0	31.0	Extens(pla)	0.525	1.749
LCCClass	13.0	67.0	NAC	0.082	0.969
CDACClass	18.0	514.0	ATMR	0.013	0.036
CIBClass	6.0	189.0	EC	0.019	0.058
H	0.062	0.333			

As métricas LCCClass, CDACClass e CIBClass foram medidas, porém não são utilizadas no modelo de avaliação da MOA4PLA. A avaliação do projeto de ALP pode ser melhor guiada utilizando métricas com grandezas semelhantes na mesma função de avaliação, pois, a influência de cada métrica na função seria semelhante. As métricas DepPack, DepIn e DepOut apresentam valores zerados, isso mostra que para as ALPs utilizadas no experimento essas métricas não tinham o que medir, visto que nenhuma das ALPs possuíam dependências de pacotes, que é o propósito da métrica DepPack, nem dependências de entrada e saída de um pacote que são o propósito de DepIn e DepOut.

A Tabela 2.2 apresenta os valores das funções de avaliação, na qual é possível notar a diferença de valores entre as funções FM(pla) e CM(pla) em relação as demais funções. Enquanto a função CM(pla) compreende valores entre 15.75 entre 31.196. A função Eleg(pla) apresenta valores entre 0.115 e 0.337.

Nos algoritmos multiobjetivos, os objetivos são analisados individualmente e independentemente dos valores totais das funções de avaliação, ou seja, a diferença de grandezas

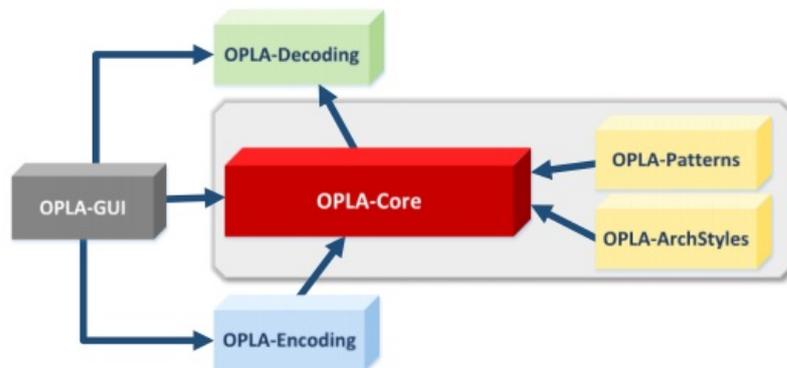
Tabela 2.2: Valores Mínimos e Máximos das Funções de Avaliação

Função de Avaliação	Valor Mínimo	Valor Máximo
CM(pla)	15.755	31.196
FM(pla)	20.735	1331.0
Ext(pla)	0.525	1.749
Eleg(pla)	0.115	0.337

entre as funções de avaliação não é um problema. Por outro lado, métricas com diferentes grandezas têm impacto maior ou menor quando agregadas em uma mesma função de avaliação.

2.3.4 Ferramenta OPLA-Tool

Para poder utilizar a abordagem MOA4PLA foi necessário desenvolver a ferramenta OPLA-Tool (Optimization for PLA Tool), proposta por Colanzi (2014) e desenvolvida em Féderle (2014). A ferramenta é composta por módulos interdependentes, como mostrado na Figura 2.5.

**Figura 2.5:** Módulos OPLA-Tool (Colanzi, 2014)

A interface com o arquiteto é a OPLA-GUI. Nela, é possível escolher qual algoritmo deverá ser utilizado para fazer a otimização, além das métricas e dos operadores de otimização. Possibilita também escolher qual ALP será utilizada como entrada para ser otimizada, além de apresentar os resultados. A OPLA-GUI utiliza os módulos OPLA-Encoding, OPLA-Decoding para fazer a codificação e decodificação da ALP de entrada, de modo que as soluções sejam legíveis para o arquiteto. De acordo com Colanzi (2014), a otimização é feita pelo OPLA-Core, que recebe uma representação de ALP da OPLA-Encoding e retorna um conjunto de ALPs resultantes para o OPLA-Decoding, que converte as soluções em uma visão legível para o arquiteto. Os módulos OPLA-Patterns

e OPLA-Arch-Styles têm o objetivo de aplicar padrões de projeto e manter as estruturas arquiteturais durante o processo de otimização.

2.3.5 Considerações Finais

Neste capítulo, foram apresentados os principais conceitos de LPS, descrevendo contextos nos quais sua adoção é adequada, bem como a sua motivação de uso. Também foram apresentadas as três atividades principais da LPS: Desenvolvimento do Núcleo de artefatos, Desenvolvimento de Produto e Gerenciamento. Igualmente foram apresentados os principais conceitos de ALP e a importância da avaliação estrutural da ALP durante o ciclo de vida da LPS.

Os principais conceitos da Engenharia de Software Baseada em Busca e de Algoritmos Genéticos também foram apresentados, bem como os conceitos mais importantes de Otimização de Arquitetura de Linha de Produto de Software, descrevendo a abordagem MOA4PLA e suas principais atividades.

Apresentaram-se, ainda, o modelo de avaliação e as métricas utilizadas, bem como as Funções Multiobjetivo: CM(pla), FM(pla), Ext(pla) e Eleg(pla). Foram também apresentadas duas tabelas: uma com a relação das métricas utilizadas, com o intuito de enfatizar a diferença de grandezas entre as métricas nas funções de avaliação; e outra com a relação das funções utilizadas no trabalho de Colanzi (2014), em que pode-se observar a diferença de grandeza entre as métricas. Diante do exposto, neste trabalho foram identificados problemas com as funções de avaliação CM(pla) e FM(pla), pois essas funções utilizam métricas com grandezas diferentes em sua avaliação da ALP. Desse modo, a proposta deste trabalho é, a partir da identificação de limitações dessas funções, obter uma avaliação mais precisa.

Assim, o próximo capítulo discute o método de pesquisa utilizado no trabalho a fim de propor um modelo de avaliação para a MOA4PLA que contenha funções que permitam medir mais precisamente as propriedades arquiteturais dos projetos de ALP.

Materiais e Método de Pesquisa

Este capítulo apresenta a metodologia utilizada neste trabalho, por meio de um diagrama de atividades e a descrição das atividades realizadas. Também apresenta os métodos de avaliação da proposta deste trabalho, que consiste em dois tipos diferentes de estudos. O primeiro foi um estudo exploratório para caracterizar o comportamento dos MOEAS na presença de múltiplos objetivos. O segundo foi um conjunto de estudos experimentais empíricos para investigar a possível correlação entre pares de funções de avaliação, as quais são propostas no Capítulo 4. Na seção referente a cada estudo, apresentam-se também as ALPs utilizadas e o método de análise dos resultados.

3.1 Sobre o Método de Pesquisa Utilizado Neste Trabalho

Para atingir o objetivo proposto, foi necessário realizar a sequência de atividades mostrada na Figura 3.1 as quais são descritas a seguir:

- Definir o tema da pesquisa: o número de métricas que compõem duas funções de avaliação na abordagem MOA4PLA, fez com que fosse proposta uma fragmentação das funções para obter uma avaliação mais precisa;
- Definir uma proposta de pesquisa: as funções de avaliação CM(pla) e FM(pla) foram separadas por propriedades arquiteturais, e seis novas funções de avaliação foram propostas, utilizando o mesmo conjunto de métricas conforme descrito no Capítulo 4;

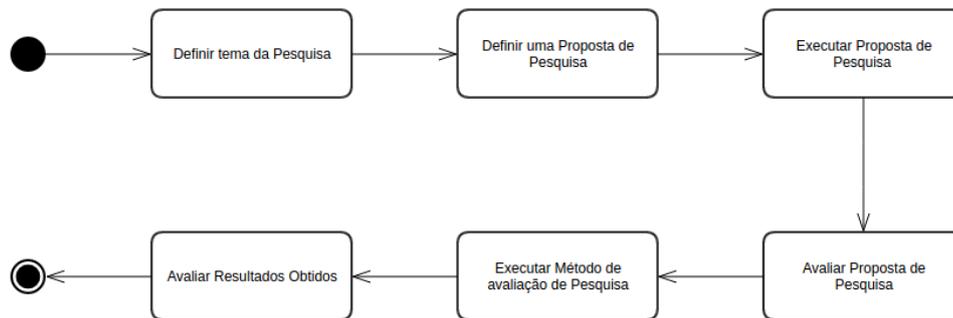


Figura 3.1: Diagrama de atividades para execução deste trabalho

- Executar a proposta de pesquisa: a execução foi feita através da implementação das funções na ferramenta OPLA-Tool que automatiza a abordagem MOA4PLA;
- Avaliar a proposta de pesquisa: a avaliação consiste em escolher o tipo de experimento e os tipos de testes a serem aplicados para fazer a avaliação das funções de avaliação propostas. Primeiramente, foi realizado um estudo exploratório para verificar o comportamento dos algoritmos na presença de múltiplos objetivos. O segundo estudo foi comparar a correlação das funções de avaliação, com a intenção de verificar se existe correlação (positiva ou negativa) entre as novas funções;
- Executar o método de avaliação da pesquisa: através de experimentos, foram realizados oito experimentos com as seis funções de avaliação propostas;
- Avaliar os resultados obtidos: por meio da análise de gráficos e testes estatísticos, é possível dizer quais funções de avaliação apresentam correlação e se elas são correlacionadas positiva ou negativamente. Além disso, foi possível avaliar se é preciso reestruturar as funções de avaliação propostas.

3.2 Avaliação da Proposta

Foram realizados dois estudos empíricos com as novas funções de avaliação. O primeiro deles trata-se de um estudo exploratório, com objetivo de avaliar o comportamento das novas funções de avaliação, em que foram utilizadas as seis novas funções de avaliação. O segundo estudo empírico engloba sete estudos experimentais. Nele, o objetivo é analisar a correlação entre as funções de avaliação propostas neste trabalho.

3.2.1 Estudo Exploratório

Inicialmente, foi realizado um estudo exploratório com as seis funções de avaliação para medir o comportamento dos MOEAs NSGA-II e PAES, durante a otimização de projetos de ALP na presença de muitos objetivos. Os MOEAs possuem algumas limitações quando aplicados a problemas com muitos objetivos, tais como: uma grande parte da população é constituída de soluções não dominadas; a avaliação da diversidade é computacionalmente cara; e o cruzamento pode ser ineficiente (Deb e Jain, 2012). Somente estudos envolvendo dois objetivos durante a otimização de projetos de ALP foram encontrados na literatura (Colanzi e Vergilio, 2014; Colanzi et al., 2014; Guizzo et al., 2014). Assim, fica evidente a importância de experimentar a aplicação de MOEAs com muitos objetivos para otimizar projetos de ALPs, já que esse problema é naturalmente influenciado por vários fatores. Sendo assim, o principal objetivo deste estudo foi avaliar o desempenho dos MOEAs NSGA-II e PAES com as seis novas funções de avaliação.

Considerando a motivação desse estudo, o seu objetivo é: analisar os conjuntos de soluções obtidos pelos MOEAs NSGA-II e PAES usando seis objetivos, com o propósito de avaliar a diversidade e o melhor custo-benefício entre estes do ponto de vista do pesquisador no contexto do projeto de ALP baseado em busca. Somente as novas funções de avaliação foram utilizadas no estudo. Dentro dele, as variáveis independentes são NSGA-II e PAES, e as dependentes são o *fitness* das soluções, formado pelos valores das seis funções de avaliação, o melhor custo-benefício entre os objetivos e a diversidade das soluções.

Abaixo segue o projeto de como foi o estudo exploratório:

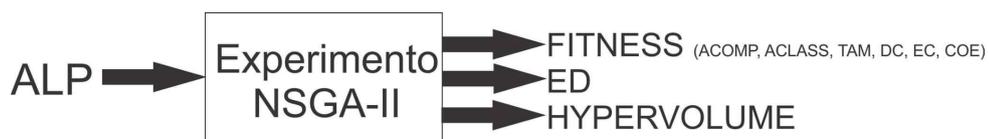


Figura 3.2: Projeto do estudo exploratório com NSGA-II

A Figura 3.2 mostra como foi projetado o estudo exploratório utilizando o algoritmo NSGA-II. Nele, o experimento recebe como entrada uma ALP, que são variáveis independentes e como variáveis dependentes as seis funções de avaliação.

A Figura 3.3 mostra como foi projetado o estudo exploratório utilizando o algoritmo PAES. Nela, o experimento recebe como entrada uma ALP, que são variáveis independentes e possuem como variáveis dependentes as seis funções de avaliação.

A hipótese nula do estudo é a hipótese H_0 , em que não existe diferença significativa entre o PAES e o NSGA-II, considerando Hipervolume (HV) e Distância Euclidiana (DE).



Figura 3.3: Projeto do estudo exploratório com PAES

As duas hipóteses alternativas do estudo são **H1** e **H2**. H1 assume que o PAES é melhor em termos de Hipervolume e Distância Euclidiana que o NSGA-II. Na hipótese H2, o NSGA-II é melhor que o PAES. Essas hipóteses são definidas abaixo em termos dos indicadores utilizados na comparação: HV e DE.

1. H0: $HV_{NSGA-II} = HV_{PAES}$ e $DENSGA-II = DE_{PAES}$;
2. H1: $HV_{NSGA-II} < HV_{PAES}$ e $DENSGA-II > DE_{PAES}$;
3. H2: $HV_{NSGA-II} > HV_{PAES}$ e $DENSGA-II < DE_{PAES}$.

A OPLA-Tool (Colanzi, 2014), ferramenta que apoia a aplicação da MOA4PLA, foi utilizada na execução do experimento, mais especificamente, o modelo OPLA-Core (Colanzi, 2014). Os MOEAs NSGA-II e PAES foram executados 25 vezes para cada ALP, e o número de rodadas foi menor que 30 vezes (número padrão) por conta de restrições de tempo. Os algoritmos foram configurados usando os mesmos parâmetros adotados por Colanzi et al. (2014). Esses parâmetros são: tamanho da população= 100, taxa de mutação = 0.9 e número de gerações = 300. Nenhum operador de cruzamento foi aplicado no NSGA-II por restrições da ferramenta.

ALPs Utilizadas no Estudo

Neste estudo foram utilizadas três ALPs, cujas principais informações são apresentadas na Tabela 3.1. Todas as ALPs são baseadas em pacotes e seguem o estilo em camadas. Arcade Game Maker (AGM) é uma LPS criada pelo SEI que inclui três jogos de arcade. Duas versões dessa ALP foram utilizadas: AGM-1 (SEI, 2015) e AGM-2 (Contieri Junior et al., 2011). A Mobile Media (MOM), versão MOM-2 (Contieri Junior et al., 2011), é uma LPS que apoia o gerenciamento de diferentes tipos de mídias. O fitness original é obtido antes da otimização na ferramenta OPLA-Tool, ou seja, são os valores antes originais de cada ALP antes da otimização. Observa-se que o valor da função de avaliação ACLASS é zero. Isso se justifica pela forma de medição das métricas CDepIn e CDepOut que contam o número de elementos dependentes de uma determinada classe e dos quais essa classe também depende, respectivamente, por meio de dependências UML ou dependências de

uso. As ALPs utilizadas não possuem esses tipos de dependências, o que justifica os valores zerados.

Tabela 3.1: Informações sobre os projetos de ALP utilizados no estudo

ALP	Compo- nentes.	Inter- faces.	Classes.	Variabi- lidades.	Caracte- rísticas.	Fitness Original (ACOMP, AClass, TAM, COE, DC e EC.
AGM-1	9	14	20	4	9	(28; 0; 35.71; 25.25; 300; 66)
AGM-2	9	14	21	5	11	(31; 0; 3.93; 29.25; 359; 112)
MOM-2	8	13	10	7	13	(27; 0; 6.153; 22.33; 405; 107)

Análise dos Resultados

Para analisar os resultados de estudo exploratório, foram utilizados alguns indicadores de qualidade: Hipervolume (HV) e Distância Euclidiana à solução ideal (DE). Na seção 2.2.3 foram usados para medir, respectivamente, a diversidade das soluções e o melhor custo benefício entre os objetivos. O teste de hipótese de Wilcoxon (Amostra única) foi utilizado com significância de 95% ($p - value \leq 0.05$), a fim de verificar a diferença estatística entre os MOEAs com relação à diversidade de soluções. Foi utilizado, também o teste de normalidade Shapiro-Wilk.

O teste de normalidade de Shapiro-Wilk (Shapiro e Wilk, 1965) calcula a variável estatística \mathbf{W} , que investiga se uma amostra é aleatória e trata de uma distribuição normal. \mathbf{W} é calculado pela seguinte equação:

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)}^2\right)}{\sum_{i=1}^n \left(x_i - \bar{x}\right)^2} \quad (3.1)$$

Abaixo a descrição da equação:

- x_i valores ordenados de amostras (x_1 o mais baixo);
- a_i constantes gerados a partir das médias, variâncias e covariâncias estatísticas de um tamanho de amostra \mathbf{n} de uma distribuição normal;

- Considerando- X uma característica do estudo, a hipótese desta experiência é:
- $H_{normal0}$: X possui distribuição normal;
- $H_{normal1}$: X não possui distribuição normal;
- Para cada W calculado há um p -value, então há evidência para refutar H_0 se p -value $>$ 0.05.

O teste de Wilcoxon é um método não-paramétrico para comparação de duas amostras pareadas. A princípio, são calculados os valores numéricos da diferença entre cada par, sendo possível três condições: aumento (+), diminuição (-) ou igualdade (=). Uma vez calculadas todas as diferenças entre os valores obtidos para cada par de dados, essas diferenças são ordenadas pelo seu valor absoluto (sem considerar o sinal), substituindo-se, então, os resultados originais pelo posto que ocupam na escala ordenada.

3.2.2 Estudos Experimentais de Correlação

Em seguida, foi realizado um estudo empírico, por meio de sete estudos experimentais com pares das funções de avaliação propostas neste trabalho. Primeiro foram realizados experimentos com pares de funções dirigidas às características; em seguida, foram testados pares de funções convencionais e, por último, pares de funções que medem a coesão em uma ALP, pois são funções que podem ser correlacionadas. Este foi, portanto, o critério para definir os pares a serem testados nos experimentos (Funções dirigidas à características, convencionais e coesão). As métricas H e LCC, da função de avaliação COE foram testadas separadamente, com a intenção de analisar se podem estar agregadas na mesma função.

Abaixo segue um exemplo de três projetos de estudos experimentais:

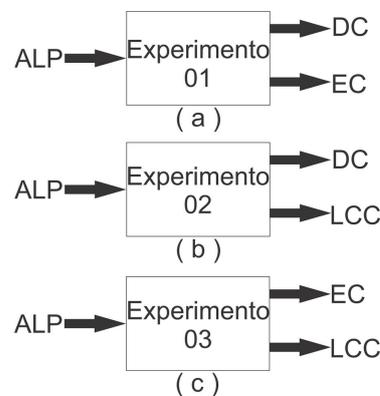


Figura 3.4: Projeto de Estudos de Correlação

A Figura 3.4 mostra como foram projetados os Experimentos 01, 02 e 03, respectivamente. Nela, cada experimento recebe como entrada uma ALP, que é variável independente e, como variáveis dependentes, tem-se as funções de avaliação.

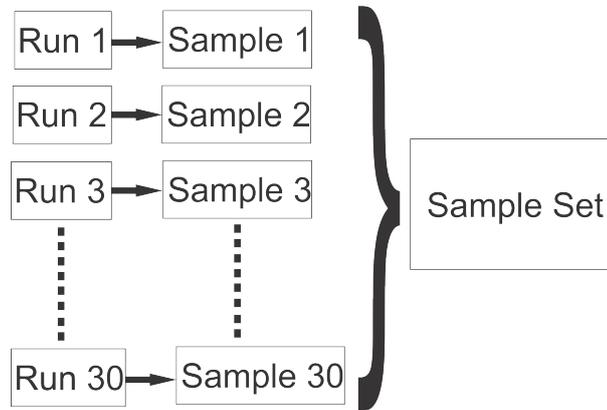


Figura 3.5: Diagramas de atividades para execução deste trabalho

A Figura 3.5 apresenta como foi executado cada experimento. Cada um dos experimentos foi repetido 30 vezes, de forma independente, para cada ALP. No final das 30 repetições, por ALP, foi possível construir um conjunto amostral de resultados. Com isso, em cada conjunto amostral foram aplicados os testes estatísticos.

Foram aplicados dois testes em cada conjunto de resultados de cada par de funções de avaliação. O primeiro teste aplicado trata-se do teste de normalidade de Shapiro-Wilk (Shapiro e Wilk, 1965). Esse teste é utilizado para determinar se um conjunto de dados (Resultados) de uma dada variável aleatória, é modelado por uma distribuição normal ou não. Como os valores apresentaram resultados não normais, aplicou-se o teste de correlação de Spearman (Hauke, 2011) com a intenção de analisar a correlação (positiva ou negativa) entre os pares de funções de avaliação.

Os experimentos foram conduzidos com as seguintes configurações na ferramenta OPLA-Tool (Féderle et al., 2015): foram executadas 30 rodadas para cada uma das ALPs, utilizando-se o MOEA NSGA-II (Deb et al., 2002), e quatro ALPs: AGM, BANK, MOM e BET. O tamanho da população foi de 100 indivíduos, o número de avaliação de *fitness* igual a 30000, e todos os operadores de mutação da MOA4PLA com uma probabilidade de mutação igual a 0.9 e sem operador de crossover. Foram utilizadas as funções ACLASS, ACOMP, TAM, H, LCC, DC e EC.

ALPs Utilizadas no Estudo

Esta seção apresenta as informações sobre as ALPs que foram utilizadas nos experimentos de correlação. A Tabela 3.2 contém informações sobre o número de características de cada ALP, ou seja, o número de interfaces que cada uma possui, assim como o número de classes e componentes de cada:

Tabela 3.2: Descrição das ALPs Utilizadas no Experimento

ALP	Componentes	Interfaces	Classes	Características
AGM	9	14	30	11
BANK	4	5	25	16
BET	56	30	115	18
MOM	8	15	14	14

A Arcade Game Maker (AGM) (SEI, 2015) é uma LPS criada pelo SEI que inclui três jogos de arcade: Brickles, Bowling e Pong. Cada jogo é praticado por um único jogador que controla, parcialmente, os objetos que se movem. O objetivo é marcar pontos acertando obstáculos estáticos. Os jogos vão desde aqueles com obstáculos baixos até obstáculos altos, e estão disponíveis para uma variedade de diferentes plataformas.

A Bank (Gomaa, 2011), por sua vez, é uma LPS que tem como finalidade gerenciar um sistema bancário, sendo possível manipular cartões, contas e caixas eletrônicos.

A BET (Donegan e Masiero, 2007), é uma LPS utilizada para gerenciar o transporte urbano, incluindo o controle de pagamentos, viagens, portões, bilhetes e outros e, por fim, a Mobile Media (MOM) (Contieri Junior et al., 2011) é uma LPS para manipular fotos, músicas e vídeos em dispositivos móveis.

Análise dos Resultados

Os sete estudos experimentais utilizaram os mesmos testes estatísticos. A fim de analisar a normalidade das amostras, utilizou-se o teste de Normalidade Shapiro-Wilk (Shapiro e Wilk, 1965), e para verificar a correlação entre as funções, foi utilizado o teste de Spearman (Hauke, 2011), que é aplicado quando a distribuição dos resultados apresenta resultados não normais. Em cada estudo experimental selecionou-se um *outlier*, e foi comparado com a PLA original. Os *outliers* foram selecionados por serem valores estatisticamente fora do padrão.

Foram utilizados os testes de normalidade, conforme descrito na Seção 3.2.1, isto é, o teste de correlação e o teste do qui quadrado, aplicados nos experimentos. Os testes de

normalidade e de correlação foram aplicados utilizando-se a ferramenta computacional **R** (Venables e Smith, 2012).

O teste de correlação de Spearman (Hauke, 2011) é o mais antigo e também o melhor método para avaliar medidas variáveis no nível ordinal, também chamado coeficiente de correlação de Spearman. Representado pela letra grega ρ , é uma medida não-paramétrica de correlação. O coeficiente de correlação (Spearman) é uma medida que apresenta as variáveis em uma escala de medição ordinal, de modo que os elementos (indivíduos ou objetos) são criados em duas séries ordenadas.

O teste de correlação de Spearman foi aplicado em cada amostra com um grau de 95% de confiança (5% nível de significância). A análise foi baseada na comparação do *p-value* obtido de cada experimento. Para $p - value > \alpha = 0.05$ não há nenhuma evidência para rejeitar H_0 , de forma que H_0 é aceito. No entanto, se $p - value < \alpha = 0.05$, então não há evidências para rejeitar H_0 e aceitar a hipótese H_1 .

- $0.00 < \rho < 0.30$, há uma correlação fraca;
- $0.30 \leq \rho < 0.60$, há uma correlação moderada;
- $0.60 \leq \rho < 0.90$, existe uma correlação forte;
- $0.90 \leq \rho < 1.00$, existe uma correlação muito forte.

Os resultados que apresentaram *outliers* receberam tratamento, no qual foram removidos 10% dos resultados com a intenção de eliminar os *outliers*. Esse tratamento consiste em eliminar ou diminuir os *outliers*. Para fazê-lo, utilizou-se o teste de qui quadrado, que é simbolizado por X^2 . Trata-se de um teste de hipóteses que se destina a encontrar um valor da divisão para duas variáveis nominais, avaliando a associação existente entre variáveis qualitativas. É um teste não paramétrico, ou seja, não depende dos parâmetros populacionais, como média e variância.

A fim de analisar o efeito das novas funções de avaliação em relação à função FM(pla) e CM(pla), foi utilizado o teste de efeito de Cliff's (Vaidyanathan, 1993). O teste de efeito é um teste que analisa a diferença entre as médias de grupos experimentais (tratamento), cada um por si, e o controle em relação à determinada medida de mudança. O campo "tratamento" refere-se aos resultados obtidos com as novas funções de avaliação, já o campo "controle" refere-se aos resultados obtidos em experimentos anteriores, utilizando as funções CM(pla) e FM(pla). O teste de efeito faz uso de uma medida da proporção relativa da variação total dos desempenhos que pode ser explicada pelo efeito da manipulação

experimental. O teste tem o objetivo de avaliar uma possível significância científica de um resultado com outro.

Este teste foi utilizado com o intenção de saber qual seria o efeito das métricas utilizadas juntas e qual efeito das métricas separadas em outras funções de avaliação. Abaixo é apresentada a escala com os níveis de efeito que cada teste pode apresentar.

O teste foi aplicado utilizando os resultados (PFtrues) obtidos nos experimentos de correlação das novas funções de avaliação e dos resultados experimentais, operando-se com as funções FM(pla) e CM(pla) (Colanzi et al., 2014). Foram utilizadas as ALPs AGM e Mobile Media (MOM). A escala de efeito empregada no teste de Cliff's segue a seguinte classificação:

- $|d| < 0.147$: Desprezível;
- $|d| < 0.33$: Pequeno;
- $|d| < 0.474$: Médio;
- **Acima:** Muito Grande.

A fim de mostrar o percentual de melhoria das novas funções de avaliação em relação ao *fitness* original, realizou-se um estudo de análise de porcentagem com os sete estudos de correlação realizados nesta dissertação. A Equação 3.2 mostra como foi calculado o percentual de melhoria. Os percentuais são comparados com as funções FM(pla) e CM(pla) (Colanzi et al., 2014). Valemo-nos do *fitness* original de cada uma das soluções dos sete estudos experimentais. Além disso, foram utilizadas as soluções de melhor e pior *fitness* de cada uma das funções objetivo de cada de um dos estudos de correlação. O *fitness* original é comparado com o *fitness* das soluções (melhor e pior).

$$Percentual = (FitnessOriginal - FitnessObtido) * 100 / FitnessOriginal \quad (3.2)$$

Foram utilizadas as ALPs AGM, Mobile Media, Bank e BET, no entanto somente no Experimento 5 foi possível utilizar a ALP Bank, pois nos demais experimentos ela não apresentou resultados com condições de comparar o percentual de melhoria.

Adotamos resultados de um estudo anterior (Colanzi et al., 2014), no qual foi feito o uso de duas versões do algoritmo NSGA-II. No entanto, neste trabalho, tomamos como referência a versão na qual todos os operadores de mutação foram utilizados, essa versão foi denominada NSGA-II-FM no trabalho de Colanzi et al. (2014).

O cálculo da porcentagem de melhoria das funções mostra o quanto cada métrica melhorou em cada uma das ALPs utilizadas nos experimentos. Se comparar esses valores com valores bases extraídos de um experimento com a função FM(pla) e CM(pla), é possível notar a diferença entre as métricas juntas em uma única função e separadas em outras funções.

3.3 Considerações Finais

Neste capítulo, tratamos da metodologia de desenvolvimento deste trabalho, descrevendo quais foram os critérios para definir o tema de pesquisa, qual o estado da arte e as contribuições esperadas. Para definir a proposta de pesquisa, foram analisadas todas funções de avaliação já implementadas, com o intuito de melhorar a avaliação das ALPs. Nesse sentido, foram implementadas as seis novas funções de avaliação a partir das funções já existentes CM(pla) e FM(pla), o que deixa a MOA4PLA agora com 10 opções de otimização, podendo o arquiteto escolher uma ou mais dentre essas opções.

Foram executados experimentos com a intenção de avaliar as novas funções de avaliação. Inicialmente, apresentou-se um estudo exploratório para avaliar o comportamento de dois MOEAs (NSGA-II e PAES) perante as seis novas funções. E, em seguida, um estudo empírico, composto de sete estudos experimentais, para saber se as funções são correlacionadas (positivamente ou negativamente).

O próximo capítulo apresenta o novo modelo de avaliação proposto para a MOA4PLA.

Novo Modelo de Avaliação Proposto para a MOA4PLA

As funções $CM(pla)$ e $FM(pla)$ envolvem métricas de diferentes propriedades arquiteturais e medidas com diferentes grandezas, o que influencia na sensibilidade da função de avaliação em apreciar o projeto de ALP alcançado no processo de busca. Resultados experimentais obtidos por Guizzo (2014) mostram que algumas soluções obtidas no processo de otimização tinham melhor acoplamento que outras, no entanto, isso não foi identificado no valor da função de avaliação $CM(pla)$. Logo, é preciso dispor de um modelo de avaliação capaz de capturar a qualidade do projeto da ALP e, por isso, é preciso refinar o modelo originalmente proposto por Colanzi (2014).

Dessa forma, este capítulo apresenta a divisão das funções $CM(pla)$ e $FM(pla)$ que envolvem propriedades arquiteturais tradicionais como acoplamento, coesão e tamanho; e propriedades de LPS como difusão e entrelaçamento de características. Essa divisão considera propriedades arquiteturais com a intenção de tentar diminuir o conjunto das métricas com diferentes grandezas. A função $CM(pla)$ utiliza métricas convencionais para medir princípios básicos de projeto como coesão e acoplamento. Na divisão, as novas funções de avaliação medem: acoplamento de pacotes $ACOMP(pla)$, acoplamento de classes $ACLASS(pla)$, tamanho de operações de uma interface $TAM(pla)$ e coesão baseada em características $COE(pla)$ que agregam as métricas LCC e H, apresentadas na Seção 2.3.2. A função $FM(pla)$ é uma agregação de varias métricas dirigidas à características. Na divisão, as novas funções de avaliação medem: difusão de características $DC(pla)$ e entrelaçamento de características $EC(pla)$. Na Tabela 2.1, a priori, as métricas contidas

na função $CM(pla)$ possuíam valores que variavam de 0.0 até 908.0. Após a divisão em novas funções de avaliação, esses valores tendem a ficar mais baixos, ou seja, as métricas medem grandezas menores dentro de uma função.

4.1 Novas Funções de Avaliação

As novas funções de avaliação propostas são apresentadas na Tabela 4.1, onde tem-se o nome de cada uma delas, juntamente com seus propósitos e as métricas utilizadas em seu desenvolvimento de cada uma das funções. Esta divisão baseia-se nas propriedades arquiteturais selecionadas e na diferença entre as grandezas das métricas apresentadas na Tabela 2.1 na qual ficou evidente a necessidade de separá-las. A métrica $DepPack$ presente no modelo original, não se encontra nas novas funções de avaliação, pois, com as ALPs utilizadas em experimentos anteriores, essa métrica apresentou valores zerados uma vez que não existe uma relação de dependência entre os pacotes, portanto não faz sentido utilizá-la.

A métrica $DepPack$ mede o número de pacotes dos quais classes e interfaces desse pacote dependem, porém, a métrica $CDepOut$ mede o número de elementos arquiteturais dos quais essa classe depende. Dessa forma, a métrica $CDepOut$ é mais completa, pois elementos arquiteturais englobam pacotes, classes e interfaces.

Tabela 4.1: Funções de Avaliação Propostas

Nome da Função	Propósito da Função de Avaliação	Métricas
$ACOMP(pla)$	Acoplamento de Componentes	$DepIn$ e $DepOut$
$ACLASS(pla)$	Acoplamento de Classes	$CDepIn$ e $CDepOut$
$TAM(pla)$	Tamanho	$NumOps$
$COE(pla)$	Coesão	H e LCC
$DC(pla)$	Difusão de Características	$CDAI$, $CDAO$ e $CDAC$
$EC(pla)$	Entrelaçamento de Características	$CIBC$, $IIBC$ e $OOBC$

A seguir, descreve-se cada uma das funções de avaliação propostas.

1. **Acoplamento de Componentes:** A Equação 4.1 é composta pelo somatório das métricas que medem as dependências de entrada $DepIn$ e de saída $DepOut$ de um pacote. Foi utilizado o somatório, pois, dessa forma, é possível somar as dependências de todos os pacotes da ALP. O objetivo da função $ACOMP(pla)$ é medir o acoplamento entre os pacotes de uma arquitetura. Na função $ACOMP(pla)$ c representa o número de pacotes da ALP:

$$ACOMP(alp) = \sum_{i=1}^c DepIn + \sum_{i=1}^c DepOut \quad (4.1)$$

2. **Acoplamento de Classes:** A Equação 4.2 é composta pelo somatório das métricas de dependência de entrada CDepIn e saída CDepOut de uma classe. Essa função mede o número de elementos de entrada e saída em que uma classe depende de outra. Também é utilizado o somatório para que seja possível obter o resultado de todas as dependências de entrada e saída de todas as classes da ALP. Na função ACLASS(pla), **c** representa o número de classes:

$$ACLASS(alp) = \sum_{i=1}^c CDepIn + \sum_{i=1}^c CDepOut \quad (4.2)$$

3. **Tamanho:** A Equação 4.3 é composta pela métrica NumOps, que mede o número de operações de uma interface. Para obter essa informação, o número total de operações é está sendo dividido pelo número de interfaces da ALP. Na função TAM(pla), **itf** representa o número de interfaces:

$$TAM(alp) = \frac{\sum_{i=1}^{itf} NumOps}{itf} \quad (4.3)$$

4. **Coesão:** A Equação 4.4 é composta pelo somatório da métrica que mede a coesão relacional entre as classes H e pelo somatório da métrica LCC, que mede a falta de coesão baseada em características dos pacotes da ALP. Na função COE(pla), **c** representa o número de pacotes:

$$COE(alp) = \sum_{i=1}^c H + \sum_{i=1}^c LCC \quad (4.4)$$

5. **Difusão de Características:** A Equação 4.5 é composta pelo somatório das métricas CDAI, CDAO e CDAC. O primeiro somatório conta o número de interfaces, que contribuem para um dado interesse. No segundo somatório, conta-se o número de operações que contribuem para a realização de um dado interesse. Por fim, no terceiro, há o número de pacotes que contribuem para a realização de um dado interesse. Na função DC(pla), **i** representa o número de interfaces e **f** o número de características de uma dada ALP:

$$DC(alp) = \sum_{i=1}^f CDAI + \sum_{i=1}^f CDAO + \sum_{i=1}^f CDAC \quad (4.5)$$

6. Entrelaçamento de Características: Esta função compõe-se pela soma do somatório das métricas que objetiva avaliar o entrelaçamento de características em uma ALP. A Equação 4.6 é composta pelo somatório das métricas CIBC, IIBC e OOBC. O primeiro somatório conta o número de interesses com os quais um dado interesse compartilha ao menos um pacote; o segundo somatório conta o número de interesses com os quais um dado interesse compartilha ao menos uma interface; e o terceiro, por fim, conta o número de interesses com os quais um dado interesse compartilha ao menos uma operação:

$$EC(alp) = \sum_{i=1}^f CIBC + \sum_{i=1}^f IIBC + \sum_{i=1}^f OOBC \quad (4.6)$$

As funções de avaliação COE(pla), DC(pla) e EC(pla) possuem, ainda, métricas com diferentes grandezas, como mostra a Tabela 2.1, porém, tratam de uma única propriedade arquitetural. Com a divisão das funções, os valores variam de dezena para centena. Antes os valores variavam de unidade a centena, por exemplo: a métrica CDAO utilizada na função DC(pla) tem o valor máximo de 908, enquanto as demais utilizadas não chegam a 200. No entanto, após o desenvolvimento das novas funções, as variações diminuíram para o intervalo de dezena para centena, o que caracteriza uma melhora nos resultados.

Dessa forma, o novo modelo de avaliação da MOA4PLA passa a ser composto por 10 funções: CM(pla), FM(pla), Ext, Eleg e as 6 novas funções propostas neste Capítulo. O arquiteto de LPS pode escolher quantos e quais objetivos deseja otimizar.

A função ACOMP(pla) tem valores zerados, pois nessas arquiteturas a função de avaliação não fez alterações, já a função ACLASS(pla) tem valores de 1.0 até 20.0, enquanto a função TAM(pla) tem valores de 4.0 até 6.66 e a função COE(pla), valores de 0.062 até 31.0.

4.2 Implementações das Funções

Para implementar as novas funções, foram necessárias alterações na ferramenta OPLA-Tool, que implicaram em mudar algumas classes, para acrescentar métodos e atributos. A seguir, explica-se quais são os elementos que foram alterados:

Foram alteradas as classes **OPLA.java** do pacote **jmetal.problems** no projeto **opla-core**, a classe **main.java** do pacote **com.ufpr.br.opla.gui2** do projeto **opla-tool** e a classe **SolutionSet.java** do pacote **jmetal.core** do projeto **architecture-representation**. Na classe **OPLA.java** foram acionados os casos (switch case) para as sete novas funções

do projeto. Ainda na classe `OPLA.java` foram implementadas as sete novas funções de avaliação multiobjetivo, operando-se com as mesmas métricas utilizadas na execução das funções já implementadas, de maneira que tais métricas são passadas como parâmetros.

Na classe `main` foram adicionadas as opções em forma de check box na tela de configurações do experimento, de modo que as sete novas funções de avaliação agora aparecem para serem selecionadas. Na classe `SolutionSet.java` foram criadas sete strings que são enviadas para o arquivo de resultado. O endereço para baixar a ferramenta é <https://github.com/OplaTool/OplaTool>.

A Figura 4.1 apresenta as classes alteradas para a implementação das funções de avaliação propostas nesta dissertação, expondo-se somente os atributos e métodos que foram adicionados ou alterados no presente trabalho. Nela os dois pacotes `core` e `problems` são relacionados ao framework `jmetal`. A classe `main` do pacote `com.ufpr.br.opla.gui2`, pertencente ao projeto `opla-tool`, faz ligação um para muitos com as classes `SolutionSet` do pacote `jmetal.core` pertencente ao projeto `architecture-representation`, e `OPLA` do pacote `jmetal.problems`, pertencente ao projeto `opla-core`. A `OPLA-Tool` utiliza classes presentes no framework para realizar as atividades. As funções de avaliação propostas foram implementadas na classe `OPLA`, aplicando-se um método para cada função:

4.3 Considerações Finais

Neste capítulo foram apresentadas as funções de avaliação propostas neste trabalho, bem como detalhes sobre a implementação destas. No próximo capítulo são apresentados os estudos experimentais realizados com o propósito de avaliar as funções de avaliação propostas.

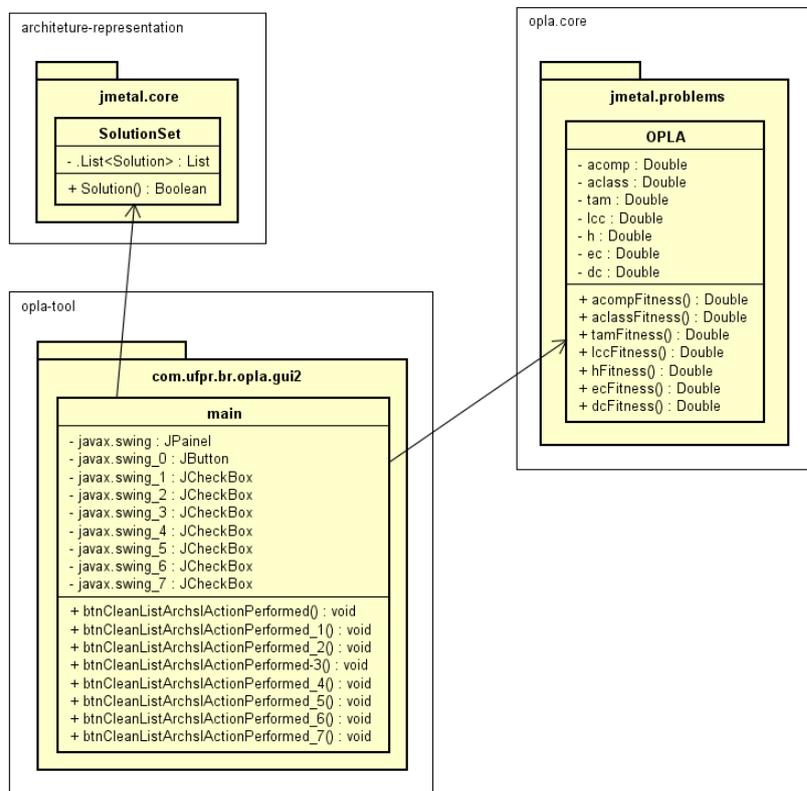


Figura 4.1: Classes Alteradas na OPLA-Tool

Estudos Experimentais

Em SBSE, a avaliação costuma ser experimental, dada a natureza estocástica dos algoritmos de busca utilizados. Dessa forma, fez-se necessário avaliar o comportamento dos MOEAs diante de múltiplos objetivos já que o modelo de avaliação da MOA4PLA é composto por dez funções de avaliação. Além disso, houve a necessidade de avaliar possíveis correlações entre as funções de avaliação, já que não é viável utilizar muitos objetivos no processo de otimização. Neste trabalho, foram realizados oito estudos empíricos, sendo um estudo exploratório com as funções seis novas funções de avaliação e sete estudos experimentais de correlação, com as funções: ACLASS, ACOMP, TAM, DC e EC, e, separadamente, com as métricas LCC e H pertencentes a função COE. Os estudos são descritos no Capítulo 3.

Esses estudos e seus resultados são descritos nas duas próximas seções.

5.1 Estudo Exploratório

Inicialmente, foi realizado um estudo exploratório para avaliar o desempenho dos MOEAs NSGA-II e PAES com as seis funções de avaliação propostas no Capítulo 4. Para analisar os resultados alcançados, foram utilizados dois indicadores de qualidade: Hipervolume e Distância Euclidiana à solução ideal. A partir deste trabalho, foi possível aprender algumas lições que permitem guiar novos estudos. O artigo completo com todos os detalhes está no Apêndice A.

5.1.1 Análise e Interpretação dos Resultados

Esta seção apresenta os resultados obtidos com a realização do experimento. A Tabela 5.1 demonstra o número de soluções que formam os conjuntos PF_{true} e PF_{known} para cada ALP. Observa-se que o NSGA-II encontrou um número menor de soluções que o PAES, porém, escolher manualmente um projeto de ALP dentre um conjunto numeroso de soluções encontradas é uma tarefa árdua para o arquiteto. Nota-se que o PF_{true} é formado por soluções encontradas pelos dois algoritmos. O NSGA-II demorou em média 10 minutos em cada execução, enquanto o PAES demorou cerca de 120 minutos.

A Tabela 5.1 apresenta o número de soluções encontradas em cada algoritmo multiobjetivo (NSGA-II e PAES), assim como o PF_{true} (Fronteira de Pareto Ótima). O NSGA-II encontrou menos soluções, portanto, foi melhor que o PAES.

Tabela 5.1: Número de Soluções Encontradas

ALP	PF_{true}	PF_{known} do NSGA-II	PF_{known} do PAES
AGM-1	471	52	423
AGM-2	811	161	657
MOM-2	696	119	577

Na Tabela 5.2 pode-se observar os valores de *fitness* e de DE (Distância Euclidiana) das soluções com menor DE encontradas pelo NSGA-II e pelo PAES, além do *fitness* da solução ideal. Os valores de *fitness* são apresentados na seguinte ordem: ACOMP, ACLASS, TAM, COE, DC e EC. Pode-se notar que as soluções encontradas pelo NSGA-II (valores destacados em negrito na tabela) são melhores que as encontradas pelo PAES, já que têm menor DE em relação à solução ideal. A solução ideal é composta pelos valores mínimos de cada função de avaliação alcançados nas soluções que compõem o PF_{true} .

Tabela 5.2: Soluções de Menor Distância Euclidiana à Solução Ideal

ALP	Solução Ideal (ACOMP, ACLASS, TAM, COE, DC e EC)	NSGA-II		PAES	
		DE	Fitness da solução	DE	Fitness da solução
AGM-1	(15; 0; 1.09; 15.1; 220; 46)	8.8	(17; 0; 3.8; 22.2; 223; 49)	226.0	(27; 0; 2.1; 43.0; 430; 124)
AGM-2	(18; 0; 1.08; 23.1; 262; 74)	8.5	(22; 0; 3.6; 27.1; 265; 79)	199.0	(24; 0; 3.2; 38.1; 455; 120)
MOM-2	(15; 0; 1.16; 17.5; 340; 70)	13.7	(19; 0; 5.7; 20.2; 345; 81)	419.6	(45; 0; 1.9; 56.0; 740; 187)

Os valores médios de hipervolume, calculados com base nos resultados de cada uma das 25 execuções de cada um dos MOEAs, podem ser vistos na Tabela 5.3. As funções objetivo foram normalizadas para o cálculo do hipervolume. Com isso, constata-se que os valores de hipervolume do NSGA-II também foram melhores do que os valores do PAES. Os maiores valores de hipervolume estão destacados em negrito.

Tabela 5.3: Hipervolume

ALP	NSGA-II		PAES		p-value
	Média	Desvio Padrão	Média	Desvio Padrão	
AGM-1	0.00338	0.00734	0.00173	0.00084	8.14507
AGM-2	0.00408	0.00031	0.00139	0.00105	9.31647
MOM-2	0.00369	0.00031	6.16777	0.00042	1.14507

Após o cálculo dos indicadores de qualidade apresentados anteriormente, o teste de normalidade Shapiro-Wilk foi aplicado sobre os resultados de HV, que aponta para o fato de os dados não seguirem uma distribuição normal. Então, foi aplicado o teste de Wilcoxon para avaliar a validade da hipótese nula adotada no presente trabalho. Os resultados evidenciam que existe diferença entre os conjuntos de dados encontrados pelos dois algoritmos com 95% de confiança (vide valores de $p - values$ na Tabela 5.3). A análise dos boxplots permite constatar a consistência dos dados obtidos pelos MOEAs, bem como qual deles obteve a maior mediana em termos de HV. Como é possível observar na Figura 5.1, o NSGA-II alcançou a maior mediana para os três conjuntos de dados. Com base na média de HV e na análise dos boxplots, observa-se que o NSGA-II encontra maior diversidade de soluções do que o PAES. Como apresentado anteriormente, NSGA-II também encontra soluções com menores DE do que PAES e, portanto, consegue otimizar mais os objetivos definidos para o processo de otimização. Desse modo, com nível de significância de 95%, a hipótese nula foi rejeitada. Ainda, o NSGA-II é melhor que PAES e, portanto, fornece indícios de que a hipótese alternativa H2 deve ser aceita, a qual afirma que o NSGA-II é melhor que PAES em termos de HV e DE. Com isso, vários pontos foram observados durante este estudo no que diz respeito à aplicação dos MOEAs com muitos objetivos. A partir dessa observação, algumas lições foram aprendidas, as quais serão abordadas na próxima seção.

5.1.2 Ameaças à validade

A principal questão que se deve levar em conta como um risco que pode afetar a validade do experimento é o tamanho da amostra e o tamanho das ALPs utilizadas, já que foram utilizados somente três projetos de ALPs acadêmicas e, com ALPs maiores, a solução

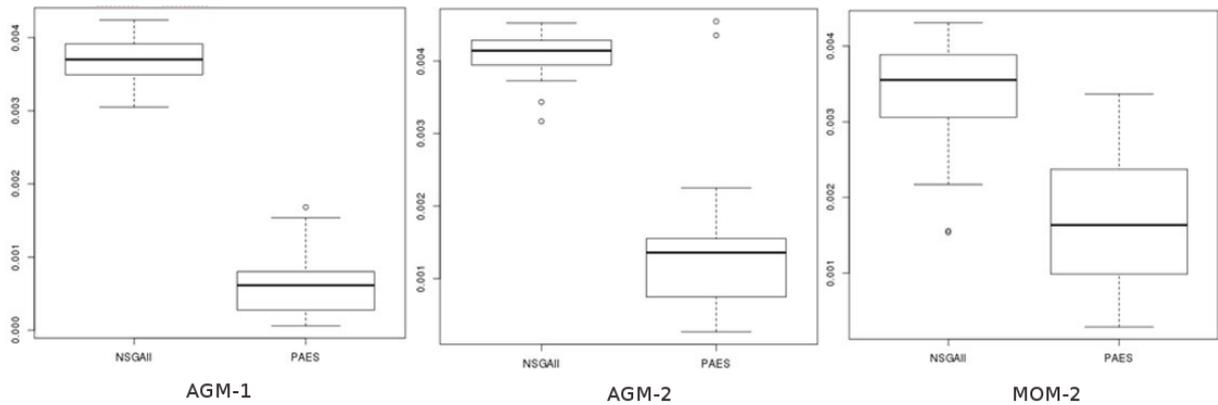


Figura 5.1: Boxplots dos Resultados de Hipervolume

do problema tende a ficar mais complexa. Todas as métricas utilizadas nas funções de avaliação e, por consequência, no experimento, já foram utilizadas em estudos anteriores (Guizzo et al., 2014). Outra possível ameaça está relacionada com a aleatoriedade dos algoritmos de busca. Para mitigar essa ameaça, cada algoritmo foi executado 25 vezes, pois isso garante resultados consistentes mesmo com resultados diferentes a cada execução. Além disso, foram adotados os mesmos parâmetros de configuração elegidos por (Colanzi e Vergilio, 2014). Todas essas ameaças podem ser tratadas em estudos futuros.

5.1.3 Discussões

A seguir, são apresentadas as lições aprendidas a partir da condução do experimento. Essas lições servem como base para a realização de futuros estudos.

Número de Soluções Encontradas: O tamanho dos conjuntos de soluções encontrados pelos dois MOEAs fornece evidências de que ambos não conseguem resolver o problema de projeto de ALP na presença de seis objetivos de forma satisfatória do ponto de vista do arquiteto, já que encontram muitas soluções, confirmando as evidências relatadas por Deb e Jain [2012]. Nesse caso, seria melhor conduzir experimentos com um número menor de objetivos.

Tempo de Execução: Considerando que o NSGA-II teve um tempo de execução médio de 10 minutos e o PAES, de 120 minutos, pode-se dizer que o tempo de execução do NSGA-II na presença de muitos objetivos é mais viável.

Otimização das Funções: Uma comparação entre os valores de *fitness* das soluções de menor DE (Tabela 5) com os valores de *fitness* originais (Tabela 3.1) permitiu observar que o NSGA-II conseguiu otimizar grande parte dos objetivos. Por exemplo, o valor original da função ACOMP para a ALP AGM-1 era 28.0. A solução de menor DE

encontrada pelo NSGA-II diminuiu o valor do acoplamento entre componentes para 17.0, enquanto o PAES somente reduziu o valor em 1 unidade (27.0). Em outros casos, como o da MOM-2, o NSGA-II conseguiu otimizar o valor enquanto o PAES obteve uma solução de pior valor que o original para essa função. O mesmo ocorreu com as funções de coesão (COE), difusão e entrelaçamento entre características (funções DC e EC). Por outro lado, o PAES conseguiu obter melhores resultados que o NSGA-II em todas as ALPs ao otimizar o tamanho das interfaces do projeto (função TAM). Considerando o acoplamento de classes (ACLASS), os algoritmos não conseguiram otimizar o valor da função, uma vez que os valores originais de todas as ALPs eram zero. Isso indica que para estas ALPs a função ACLASS não tinha o que otimizar.

Soluções Próximas do Ideal: O NSGA-II conseguiu encontrar soluções próximas da solução ideal mesmo diante do desafio de otimizar seis objetivos simultaneamente.

Número de Funções: Apesar de usar seis objetivos, o NSGA-II conseguiu valores de *fitness* melhores que os originais, como mostram as tabelas 5.1, 5.2 e 5.3. Isso evidencia tanto a viabilidade de aplicação das funções propostas como a sua sensibilidade às mudanças nas propriedades arquiteturais que elas pretendem medir.

Os resultados fornecem indícios de que NSGA-II supera PAES para resolver problemas com seis objetivos, considerando as ALPs utilizadas. No entanto, o desempenho do NSGA-II ainda assim não é adequado, pois produz uma grande quantidade de soluções, dificultando a seleção de um projeto de ALP por parte do arquiteto, ou seja, ele não consegue explorar adequadamente o espaço de busca diante de muitos objetivos. Nesse sentido, pode-se concluir que é viável a utilização das novas funções propostas neste trabalho, porém, seria melhor utilizá-las em conjuntos variados de funções de avaliação, operando com um menor número de objetivos.

5.2 Estudos de Correlação

Foi realizado um estudo empírico, composto por sete estudos experimentais. Nesses estudos, é analisada a correlação entre as funções de avaliação: ACLASS, ACOMP, TAM, COE, DC e EC, e, separadamente, com as métricas LCC e H pertencentes à função COE. Os estudos foram planejados de acordo com a Tabela 5.4. Na tabela são apresentados os nomes dos experimentos executados, o objetivo de cada estudo experimental, quais funções e métricas foram utilizadas e, na última coluna, são apresentados quais dados foram calculados para serem utilizados na análise quantitativa dos resultados obtidos. Em todos os experimentos, fez-se uso do algoritmo NSGA-II e a mutação foi ajustada em 0.9 na ferramenta OPLA-Tool. Os dados coletados foram o *Fitness* das soluções em

todos os experimentos. Esses experimentos foram conduzidos para responder se existe correlação entre as novas funções de avaliação propostas neste trabalho.

O alvo desses estudos experimentais foi investigar a correlação significativa entre combinações das funções (DC e EC), (DC e LCC), (EC e LCC), (ACOMP e TAM), (ACCLASS e ACOMP) e (ACOMP e H). Realizou-se, também, um estudo sobre a função COE, avaliando a correlação entre as métricas (LCC e H) que compõem a função.

Foram utilizados quatro projetos de ALPs: AGM, MOM, BANK e BET. Cada estudo envolveu um par de funções de avaliação.

As próximas seções apresentam os sete estudos de correlação, bem como mostram, em cada uma delas, a hipótese nula (H_0) na qual não existe correlação significativa entre as funções de avaliação, e (H_1), em que existe correlação significativa entre as funções de avaliação. Também apresenta os resultados do teste de normalidade aplicado nos conjuntos de soluções obtidos, e as informações são apresentadas através de uma tabela que contém o p-value e a distribuição (normal ou não normal) para cada ALP em cada uma das funções de avaliação. São apresentados os resultados do teste de correlação entre as funções, através de tabelas que contém o p-value o nível de correlação e a conclusão sobre a correlação. São apresentados os gráficos de boxplots para cada uma das ALPs e para cada uma das funções de avaliação. Por último, apresenta-se uma discussão sobre a correlação das funções de avaliação.

A Tabela 5.5 apresenta o número médio de soluções obtidas em cada um dos sete estudos experimentais de correlação. A título de ilustração o Apêndice B apresenta uma solução obtida para a ALP AGM em cada um dos sete estudos experimentais. Foram utilizadas uma solução boa de cada um dos sete experimentos.

Tabela 5.5: Número Médio de Soluções Encontradas por Experimento

PLA	Experimentos						
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7
AGM	3	8	41	12	15	1	1
BANK	1	1	1	1	3	1	1
BET	3	3	10	11	21	4	3
MOM	7	3	18	3	23	1	2

Com exceção do Experimento 5, todos os experimentos retornaram somente uma solução para a ALP BANK, inviabilizando a aplicação dos testes estatísticos. Quando somente uma solução é retornada, pode-se dizer que o NSGA-II não conseguiu otimizar o projeto de ALP original; ou porque já era um bom projeto ou porque o algoritmo encontrou a solução ótima.

Tabela 5.4: Estudos Experimentais das Funções de Avaliação

Experi- mento	Objetivo	Funções de Avaliação	Métricas	Dados Coletados	Dados Calculados
1	Analisar a correlação entre as funções DCxEC	DC e EC	X	Fitness da Solução	Correlação de DCxEC
2	Analisar a correlação entre as funções DCxLCC	DC e LCC	X	Fitness da Solução	Correlação de DCxLCC
3	Analisar a correlação entre as funções ECxLCC	EC e LCC	X	Fitness da Solução	Correlação de LCCxEC
4	Analisar a correlação entre as funções HxLCC	X	HxLCC	Fitness da Solução	Correlação de HxLCC
5	Analisar a correlação entre as funções ACOMPx TAM	ACOMP e TAM	X	Fitness da Solução	Correlação de ACLASSx TAM
6	Analisar a correlação entre as funções ACLASSx ACOMP	ACLASS e ACOMP	X	Fitness da Solução	Correlação de ACLASSx ACOMP
7	Analisar a correlação entre as funções COEx ACOMP	ACOMP	H	Fitness da Solução	Correlação de HxACOMP

As Tabelas 5.6 e 5.7 apresentam o *fitness* original das ALPs utilizadas em cada um dos sete estudos de correlação realizados nesta seção. O *fitness* original é o valor original das funções para as ALPs, ou seja, é o valor das funções para as ALPs antes da otimização. O *fitness* original é utilizado para analisar a correlação em cada um dos pares de funções para cada ALP.

Tabela 5.6: Tabela com *Fitness* Original Experimentos de Correlação - Parte 1

Fitness Original								
ALP	Experimento1		Experimento2		Experimento3		Experimento4	
AGM	Dc	430	Dc	430	Ec	124	Lcc	26
	Ec	124	Lcc	26	Lcc	26	H	7
BANK	Dc	146	Dc	146	Ec	130	Lcc	16
	Ec	130	Lcc	16	Lcc	16	H	2.0
MOM	Dc	680	Dc	680	Ec	225	Lcc	28
	Ec	225	Lcc	28	Lcc	28	H	7
BET	Dc	688	Dc	688	Ec	200	Lcc	100
	Ec	200	Lcc	100	Lcc	100	H	88

Tabela 5.7: Tabela com *Fitness* Original Experimentos de Correlação - Parte 2

Fitness Original						
ALP	Experimento5		Experimento6		Experimento7	
AGM	Acomp	3.928	Aclass	30	Acomp	3.928
	Tam	59	Acomp	3.928	H	7
BANK	Acomp	1.8	Aclass	133	Acomp	1.8
	Tam	24	Acomp	1.8	H	2
MOM	Acomp	5.866	Aclass	14	Acomp	5.866
	Tam	58	Acomp	5.866	H	7
BET	Acomp	5.4	Aclass	122	Acomp	5.4
	Tam	339	Acomp	5.4	H	48

As soluções de melhor *trade-off* entre os objetivos utilizados em cada estudo experimental é apresentada no Apêndice A. As próximas seções apresentam os resultados de cada experimento.

5.2.1 Experimento 1

No primeiro experimento é analisada a correlação entre as funções de avaliação DC e EC. As variáveis independentes são as ALPs utilizadas, enquanto as funções DC e EC são as variáveis dependentes.

Hipótese

A hipótese nula é a hipótese H_0 , na qual não existe correlação significativa entre as funções de avaliação, enquanto em H_1 em que existe correlação significativa entre as funções de avaliação.

- H_{dc-ec0} : Não existe uma correlação significativa entre DC e EC ;
- H_{dc-ec1} : Há correlação significativa entre DC e EC .

Resultados

A Tabela 5.8 apresenta os resultados dos testes de normalidade aplicados no Experimento 1. Somente a ALP BET na função DC apresentou distribuição normal, como todos os demais conjuntos de soluções têm distribuição não normal, utilizou-se o teste de correlação de Spearman.

Tabela 5.8: Experimento 1 - Teste de Normalidade - Funções DC x EC

PLA	Funções de Avaliação			
	DC		EC	
	p-value	Distribuição	p-value	Distribuição
AGM	0.02592	Não normal	0.00009873	Não normal
BET	0.2098	Normal	0.00000000001084	Não normal
MOM	0.0005665	Não normal	0.000000000427	Não normal

A Tabela 5.9 apresenta os resultados de testes de correlação aplicados no Experimento 1. O teste da ALP AGM com as funções de avaliação DC e EC apresentou $p\text{-value} > 0.05$, o que implica que não há correlação entre estas funções, isto é, considerando o grau de correlação, existe uma correlação significativa entre as funções DC e EC. Para as ALPs BET e MOM existe forte correlação negativa, a AGM não apresentou correlação, pois o valor do $p\text{-value}$ foi maior que 0.05.

Tabela 5.9: Experimento 1 - Teste de Correlação - Funções DC x EC

PLA	Teste Aplicado	Resultado		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.455	-0.08527043	Não Existe Correlação
BET	Spearman	0.000000000000000022	-0.7188568	Existe Correlação Forte Negativa
MOM	Spearman	0.000000000000000022	-0.731903	Existe Correlação Forte Negativa

A Figura 5.2 apresenta os boxplots da ALP AGM. A análise do boxplot complementa a análise de correlação para as funções DC e EC. Para DC, os resultados obtidos foram: mínimo de 243 e máximo de 369. Com a mediana perto de 320, pode-se dizer que os dados são simétricos. Na função EC, os valores obtidos foram: mínimos de 62 e máxima de 96, no entanto, uma vez que a mediana é 80 e não são equidistantes a partir da caixa extremos, pode-se dizer que os dados são assimétricos. Isso significa que enquanto os valores de DC foram otimizados entre 243 e 369, os valores de EC não foram otimizados de forma simétrica em relação ao mesmo. O gráfico da função EC apresenta ocorrência de *outliers*. Devido à ocorrência de *outliers* para a função EC, foi feito um tratamento do mesmo, conforme descrito no Capítulo 3.2.2. Entretanto, mesmo com o tratamento, esse boxplot ainda apresenta ocorrência de *outliers*.

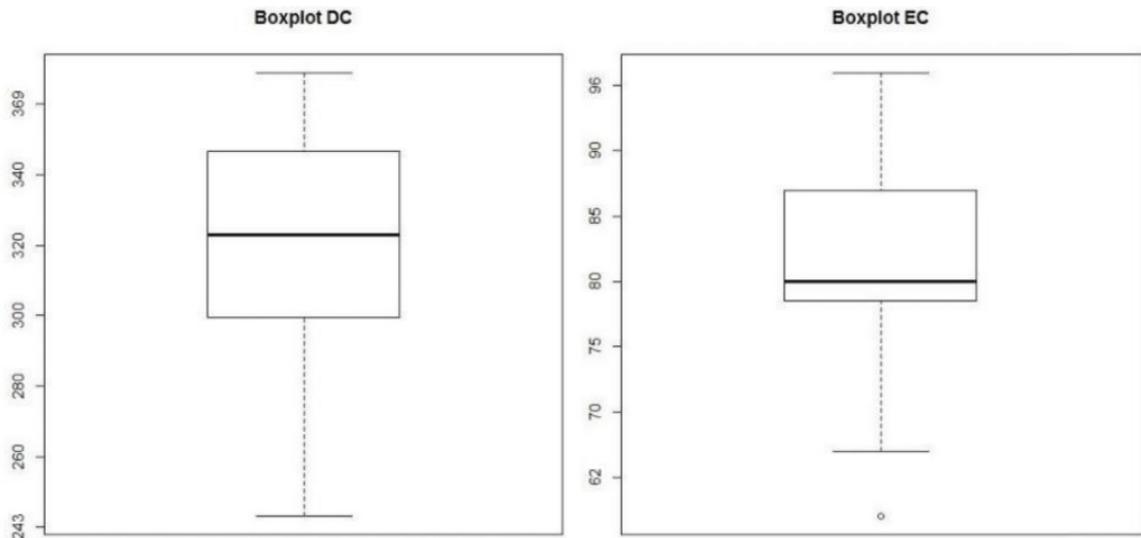


Figura 5.2: Experimento 1 - Boxplots DC e EC (AGM)

A análise do boxplot para as funções DC e EC para a ALP BET é apresentada na Figura 5.3. Esses resultados mostram que a função DC tem uma gama de valores compreendidos entre 434 e 611, com a mediana entre 500 e 550, não sendo equidistante das extremidades do boxplot e, portanto, os dados apresentados são assimétricos. A função EC retorna uma variação de cerca de 100 unidades, com valores entre 137 e 235. A mediana encontra-se perto da extremidade superior do boxplot, de forma que os dados são assimétricos. Ou seja, a grande maioria dos resultados fica bem perto do valor original. Uma possível explicação para isso seria o fato do projeto original ser bom (poucas características entrelaçadas), enquanto em DC há várias alternativas para otimização.

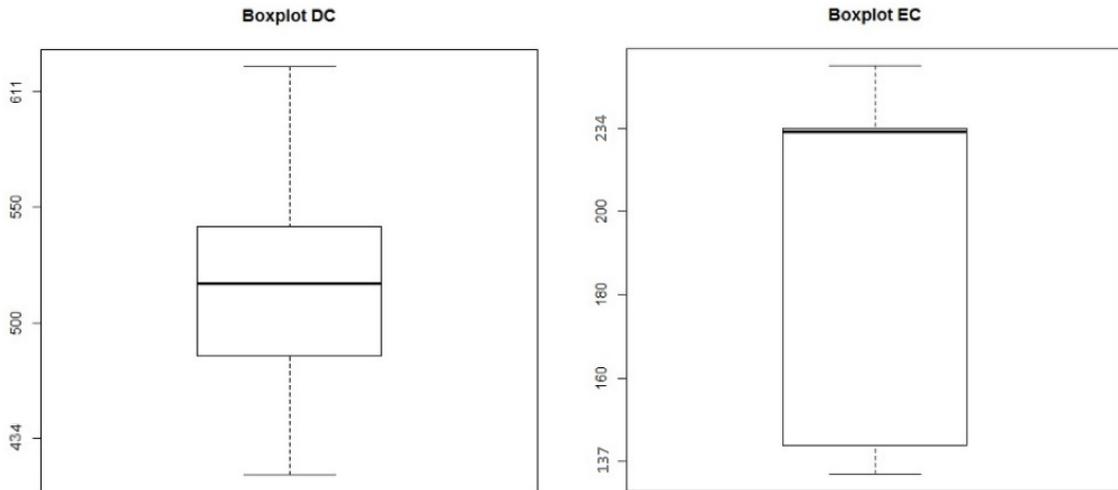


Figura 5.3: Experimento 1 - Boxplots DC e EC (BET)

Finalmente, os boxplots DC e EC da ALP MOM são representados na Figura 5.4. A função DC demonstra *outliers* e intervalos de valores entre o mínimo de 496 e o máximo de 565; com a mediana entre 520 e 530, equidistante do quarto quartil, portanto pode-se dizer que os dados são simétricos. Analisando os resultados da função EC, que está na gama de 104-161, foram encontrados quatro *outliers*, bem como a mediana está perto de 120 e não é equidistante das extremidades do boxplots. Assim, entende-se que os resultados são assimétricos. Devido à ocorrência de *outliers*, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, esse boxplot ainda apresenta tal ocorrência.

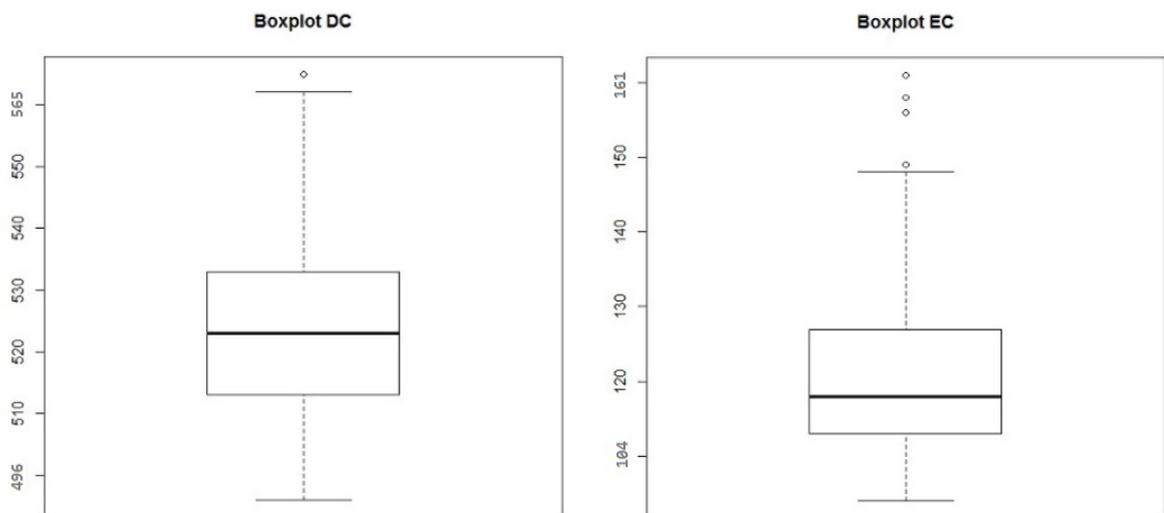


Figura 5.4: Experimento 1 - Boxplots DC e EC (MOM)

Discussões considerando a ALP MOM e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (DC;EC), é (690;150), e a solução original de *fitness* igual a (680;225), é possível observar que, na solução *outlier*, o número de características que se entrelaçam com uma característica específica é maior do que na solução original. Isso justifica a diferença no valor de *fitness* referente à EC. No entanto, uma característica é realizada em menos elementos (classes, componentes e interfaces) na solução *outlier* em relação ao *fitness* original. As características estão mais difusas na solução original e no *outlier* em relação ao seu entrelaçamento de características, porém, no *outlier* essa diferença é maior.

É possível observar a correlação negativa entre as funções, pois enquanto o valor de DC diminui, o de EC aumenta. Esse comportamento leva em consideração a comparação entre o *outlier* e a original.

Dessa forma, há evidências para refutar H_{dc-ec0} e aceitar H_{dc-ec1} . Assim, entende-se que houve correlação negativa significativa forte entre as métricas DC e EC.

5.2.2 Experimento 2

No experimento 2 é analisada a correlação entre as funções de avaliação DC e LCC. As variáveis independentes são as ALPs utilizadas, ao passo que as funções DC e LCC são as variáveis dependentes.

Hipótese

A hipótese nula é hipótese H_0 , na qual não existe uma correlação significativa entre as funções de avaliação, e H_1 , em que existe correlação significativa entre as funções de avaliação.

- $H_{dc-lcc0}$: Não existe uma correlação significativa entre *DC* e *LCC*;
- $H_{dc-lcc1}$: Há correlação significativa entre *DC* e *LCC*.

Resultados

A Tabela 5.10 apresenta os resultados dos testes de normalidade aplicados no Experimento 2. Todos os resultados demonstram soluções não normais, portanto foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.10: Experimento 2 - Teste de Normalidade - Funções DC x LCC

PLA	Funções de Avaliação			
	DC		LCC	
	p-value	Distribuição	p-value	Distribuição
AGM	0.00000000009315	Não normal	0.00000003112	Não normal
BET	0.04045	Não normal	0.000000007567	Não normal
MOM	0.000000002709	Não normal	0.000000001072	Não normal

A Tabela 5.11 apresenta os resultados dos testes de correlação do Experimento 2. O teste mostrou que, para todos os projetos de ALPs, retornou-se a $p - value < 0.05$, onde existe correlação forte negativa.

Tabela 5.11: Experimento 02 - Teste de Correlação - Funções DC x LCC

PLA	Teste Aplicado	Resultados		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.000000000000000022	-0.865123	Existe Correlação Forte Negativa
BET	Spearman	0.000000000000003316	-0.7206745	Existe Correlação Forte Negativa
MOM	Spearman	0.000000000000000022	-1	Existe Correlação Muito Forte Negativa

A Figura 5.5 mostra os gráficos de boxplots das funções DC e LCC para a ALP AGM. Os valores da função DC estão entre o mínimo de 252 e o máximo de 460, com uma mediana de 350, e não são equidistantes das extremidades do boxplot, portanto, os dados são assimétricos. Nesse experimento, a função DC apresentou *outliers* acima do terceiro quartil e abaixo do primeiro quartil. Os dados de LCC mostraram valores mínimos igual a 24 e no máximo igual a 39, com a mediana perto de 30, bem como sendo equidistantes a partir dos extremos, de modo que é possível dizer que os valores são simétricos. Devido à ocorrência de *outliers* para a função EC, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, neste boxplot ainda há *outliers*.



Figura 5.5: Experimento 2 - Boxplots DC e LCC (AGM)

A Figura 5.6 mostra os gráficos de boxplots para as funções DC e LCC para a ALP BET. A função DC exibe um *outlier* acima do primeiro quartil e uma gama de valores entre: 441 e 624, a mediana perto de 500. Dessa forma, conclui-se que os dados são assimétricos. A função LCC apresenta uma gama de variação entre os valores: 137 e 241, com a mediana perto de 220 e perto do terceiro quartil, caracterizando os dados como assimétricos. Como a função DC apresentou *outlier*, fez-se um tratamento do mesmo, conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresenta ocorrência de *outliers*.

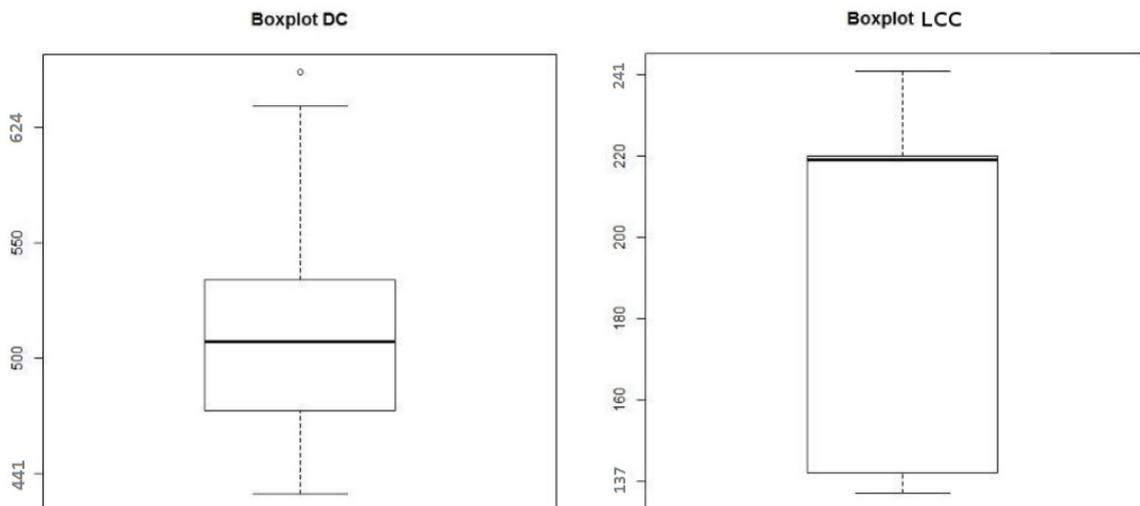


Figura 5.6: Experimento 2 - Boxplots DC e LCC (BET)

A Figura 5.7 exibe os gráficos de boxplots para as funções DC e LCC, para a ALP MOM. Nos gráficos em cada uma das 30 (trinta) rodadas, foram geradas apenas três soluções e todas as funções possuem os mesmos valores para as três soluções encontradas. Os valores da função DC são de 636, 653 e 680. Na função LCC, os valores são de 28, 30 e 31. Em ambos os boxplots a mediana apresentada não é equidistante das extremidades do boxplot e, portanto, pode-se concluir que em ambos os casos os dados são assimétricos. Devido à ocorrência de *outliers* em ambas as funções, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, estes boxplots ainda apresentam *outliers*.

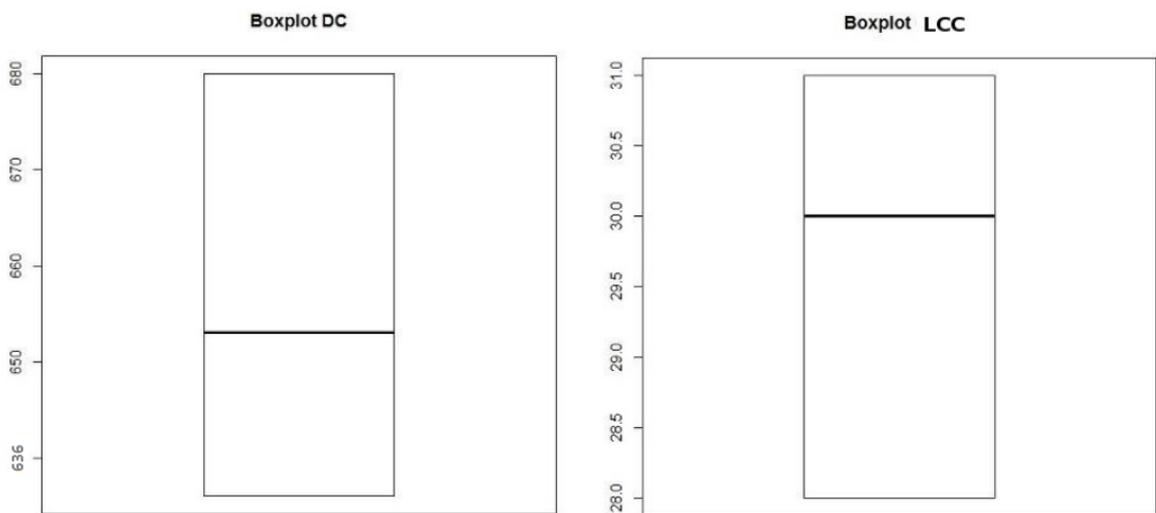


Figura 5.7: Experimento 2 - Boxplots DC e LCC (MOM)

Discussões considerando a ALP AGM e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (DC;LCC), é (252;37), e a solução original de *fitness* igual a (430;26), pode-se observar que na solução original existem mais elementos (classes, componentes e interfaces) para implementar uma característica, isso justifica o valor do *fitness* original em relação ao *outlier*. Ao mesmo tempo em que a falta de coesão baseada em características dos componentes é maior no *outlier*, a difusão de características é maior tanto no *outlier* quanto na solução original, pois em ambas DC é maior. É possível observar a correlação negativa entre as funções, pois enquanto o valor de DC diminui, o de LCC aumenta. Esse comportamento leva em consideração a comparação entre o *outlier* e a original.

Nesse sentido, há evidências para refutar $H_{dc-lcc0}$ e aceitar $H_{dc-lcc1}$. Com isso, entende-se que houve correlação negativa significativa forte entre as métricas DC e LCC.

5.2.3 Experimento 3

No terceiro experimento é analisada a correlação entre as funções EC e LCC. As variáveis independentes são as ALPs utilizadas, ao passo que as funções EC e LCC são as variáveis dependentes.

Hipótese

A hipótese nula é a H_0 , na qual não existe uma correlação significativa entre as funções de avaliação, e H_1 , em que existe correlação significativa entre as funções de avaliação.

- $H_{ec-lcc0}$: Não existe uma correlação significativa entre EC e LCC ;
- $H_{ec-lcc1}$: Há correlação significativa entre EC e LCC .

Resultados

A Tabela 5.12 apresenta os resultados dos testes de normalidade aplicados no Experimento 3. Todos os valores demonstram resultados não normais, de maneira que foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.12: Experimento 3 - Teste de Normalidade - Funções EC x LCC

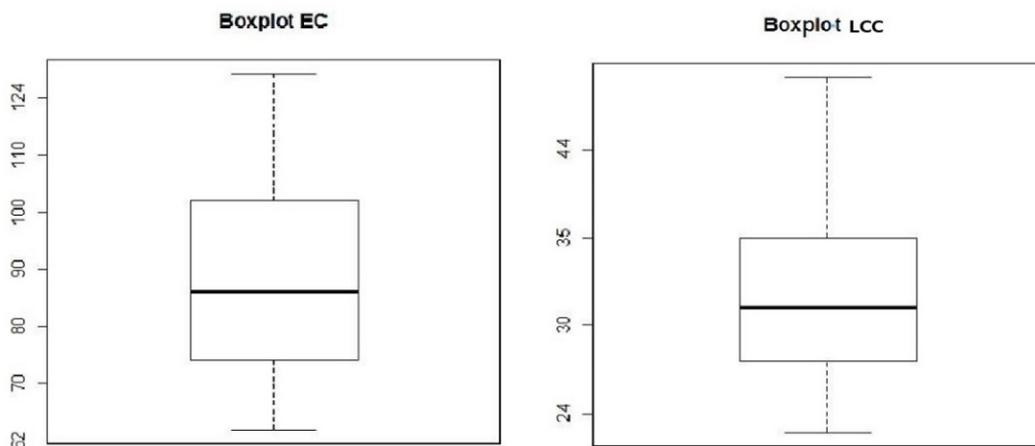
PLA	Funções de Avaliação			
	EC		LCC	
	p-value	Distribuição	p-value	Distribuição
AGM	0.000000000000000022	Não normal	0.000000000000000022	Não normal
BET	0.000000000000000022	Não normal	0.000000000000000022	Não normal
MOM	0.000000000000000022	Não normal	0.0000000000006012	Não normal

Os resultados do teste de correlação aplicado para o Experimento 3 estão exibidos na Tabela 5.13. Para as três ALPs, o teste retornou $p - value < 0.05$, indicando correlação significativamente negativa forte em todos as ALPs.

Tabela 5.13: Experimento 3 - Teste de Correlação - Funções EC x LCC

ALP	Teste Aplicado	Resultados		
		p-value	Nível de Correção	Conclusão
AGM	Spearman	0.00000000000000022	-0.896511	Existe Correlação Forte Negativa
BET	Spearman	0.00000000000000022	-0.9969927	Existe Correlação Muito Forte Negativa
MOM	Spearman	0.00000000000000022	-0.7599877	Existe Correlação Forte Negativa

A Figura 5.8 apresenta os boxplots das funções EC e LCC com a ALP AGM. A função EC mostrou uma gama de valores de dados diferentes, divergindo entre um mínimo de 62 e um máximo de 124. A mediana com valor igual a 85 revela que os dados não são equidistantes dos extremos do boxplot, portanto, pode-se concluir que os dados são assimétricos. A função LCC não demonstra uma grande variação nos valores e apresenta valores mínimos e máximos de 24 e 44, respectivamente, com a mediana perto de 30, sendo os dados assimétricos. Devido à ocorrência de *outliers* na função DC, foi feito um tratamento do mesmo, conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresenta *outliers*.

**Figura 5.8:** Experimento 3 - Boxplots EC e LCC (AGM)

Os valores das funções EC e LCC para a ALP BET são mostrados na Figura 5.9. A função EC apresentou um intervalo entre 183 e 256 com mediana próxima do terceiro quartil e valor 230, revelando a assimetria dos dados. A função LCC, por sua vez, apresenta um intervalo entre 75 e 95, com mediana próxima do primeiro quartil com valor 80, fato que também demonstra a assimetria dos dados.

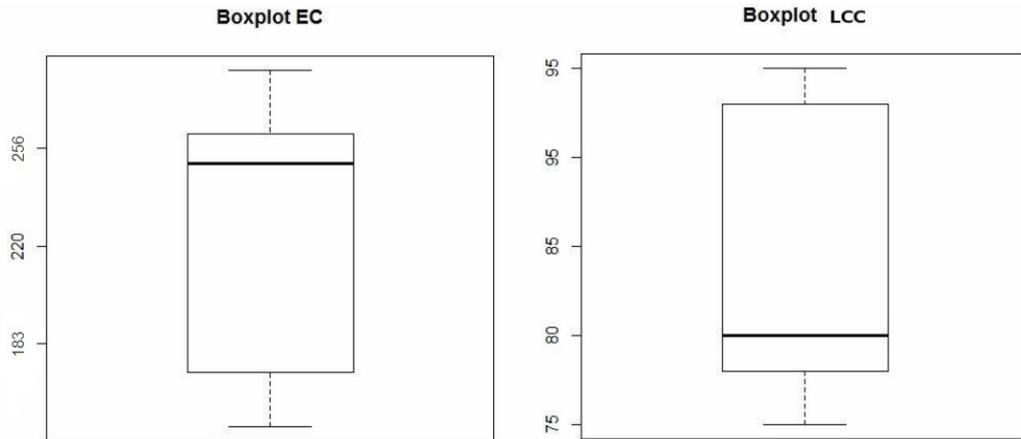


Figura 5.9: Experimento 3 - Boxplots EC e LCC (BET)

A Figura 5.10 mostra os gráficos do boxplot das funções EC e LCC para a ALP MOM. A função EC apresenta doze *outliers*. O valor da função EC está entre 104 e 186 com mediana perto de 31. Uma vez que a mediana não é equidistante das extremidades do boxplot, entende-se que os dados são assimétricos para as duas funções. Devido à ocorrência de *outliers* em ambas as função, foi feito um tratamento destes, conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, estes boxplots ainda apresentam de *outliers*.

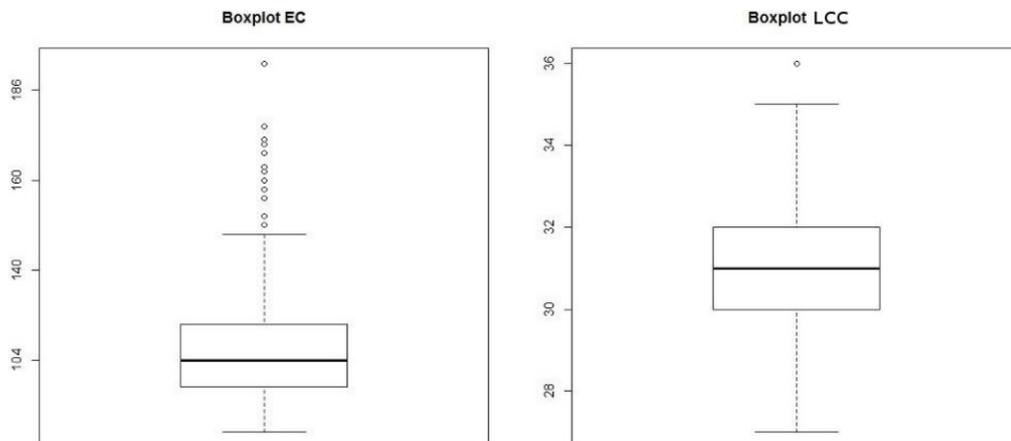


Figura 5.10: Experimento 3 - Boxplots EC e LCC (MOM)

Discussões

Considerando a ALP MOM e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (EC;LCC), é (162;30) e a solução original de *fitness* igual a (225;28), a função EC apresentou um valor mais baixo em relação ao *fitness* original, enquanto LCC aumentou. Esses valores se justificam, pois, quanto menor forem os valores de IIBC, OIBC e CIBC (EC), menor é o entrelaçamento entre características. O entrelaçamento de características (EC) é maior que a falta de coesão baseada em características (LCC), tanto no *outlier* quanto na solução original.

É possível observar a correlação negativa entre as funções, pois enquanto o valor de EC aumenta, o de LCC diminui. Esse comportamento leva em consideração a comparação entre o *outlier* e a original.

Dessa forma, há evidências para refutar $H_{ec-lcc0}$ e aceitar $H_{ec-lcc1}$. Logo, entende-se que houve correlação negativa significativa forte entre as métricas EC e LCC.

5.2.4 Experimento 4

No quarto experimento é analisada a correlação entre as métricas H e LCC, que compõem a função de avaliação COE proposta, proposta neste trabalho. Cada uma das métricas constitui uma função de avaliação neste experimento a fim de averiguar se elas são correlacionadas ao ponto de poderem ser agregadas em uma mesma função de avaliação, como proposto na Seção 4. Dessa forma, cada métrica será tratada como uma função neste experimento. As variáveis independentes são as ALPs utilizadas, e as métricas são as variáveis dependentes.

Hipótese

A hipótese nula é hipótese H_0 , na qual não existe uma correlação significativa entre as métricas, e H_1 , em que existe correlação significativa entre as métricas da função de avaliação COE.

- H_{h-lcc0} : Não existe uma correlação significativa entre H e LCC ;
- H_{h-lcc1} : Há correlação significativa entre H e LCC .

Resultados

A Tabela 5.14 apresenta os resultados dos testes de normalidade aplicados no Experimento 4. Todos os resultados demonstram soluções não normais, portanto foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.14: Experimento 4 - Teste de Normalidade - Métricas H x LCC

PLA	Métricas Avaliadas			
	H		LCC	
	p-value	Distribuição	p-value	Distribuição
AGM	0.000000000004868	Não normal	0.0000000137	Não normal
BET	0.000000000000000022	Não normal	0.000000004794	Não normal
MOM	0.00000005816	Não normal	0.000000000001358	Não normal

Os resultados do teste de correlação aplicado para o Experimento 4 estão apresentados na Tabela 5.15. Para as três ALPs, o teste retornou $p - value < 0.05$ e os valores de nível de correlação negativa significativamente forte.

Tabela 5.15: Experimento 4 - Teste de Correlação - Métricas LCC x H

ALP	Teste Aplicado	Resultados		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.000000000000000022	-0.8678138	Existe Correlação Forte Negativa
BET	Spearman	0.000000000000000022	-0.9607249	Existe Correlação Muito Forte Negativa
MOM	Spearman	0.000000000000000022	-0.7887382	Existe Correlação Forte Negativa

A Figura 5.11 apresenta os boxplots das funções H e LCC com a ALP AGM. A função LCC mostrou uma gama de valores divergindo entre um mínimo de 24 e máximo de 45. A mediana perto de 30 indicando que os valores não são equidistantes dos extremos do boxplot, portanto, pode-se concluir que os dados são simétricos. No entanto, a função H apresenta uma variação maior nos valores. Além disso, revela valores mínimos e máximos de 550 e 750, respectivamente, demonstrando uma mediana de 650, de modo a implicar em dados assimétricos.

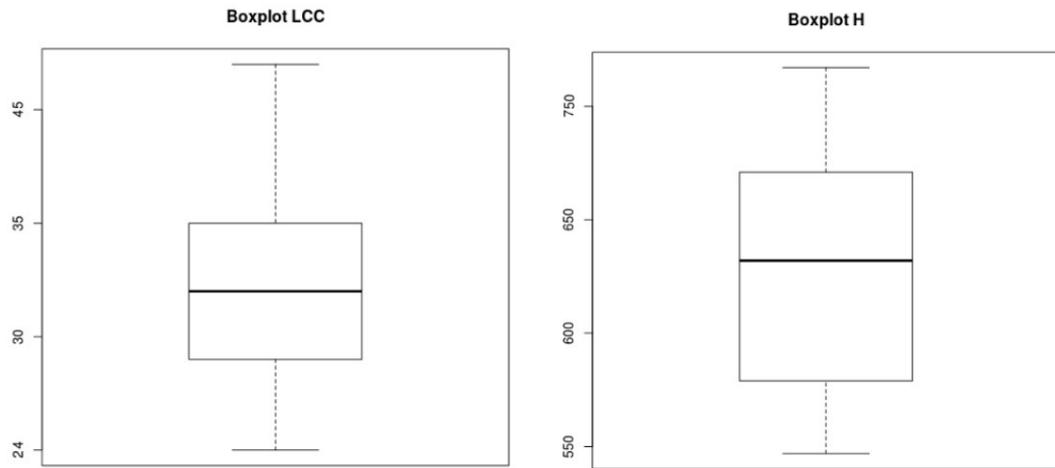


Figura 5.11: Experimento 4 - Boxplots LCC e H (AGM)

Os boxplots das funções LCC e H para a ALP BET aparecem na Figura 5.12. A função H apresentou um intervalo pequeno entre 70 e 95. A mediana com valor próximo de 85 indica que os dados são simétricos. A função LCC exibiu um intervalo maior entre 1300 e 1900, com valores maiores que a métrica LCC. A mediana com valor de 1400, indica que os valores são assimétricos. Além disso, a função LCC apresenta *outliers*, bem acima de terceiro quartil. Devido à ocorrência de *outliers* na função LCC, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresenta *outliers*.

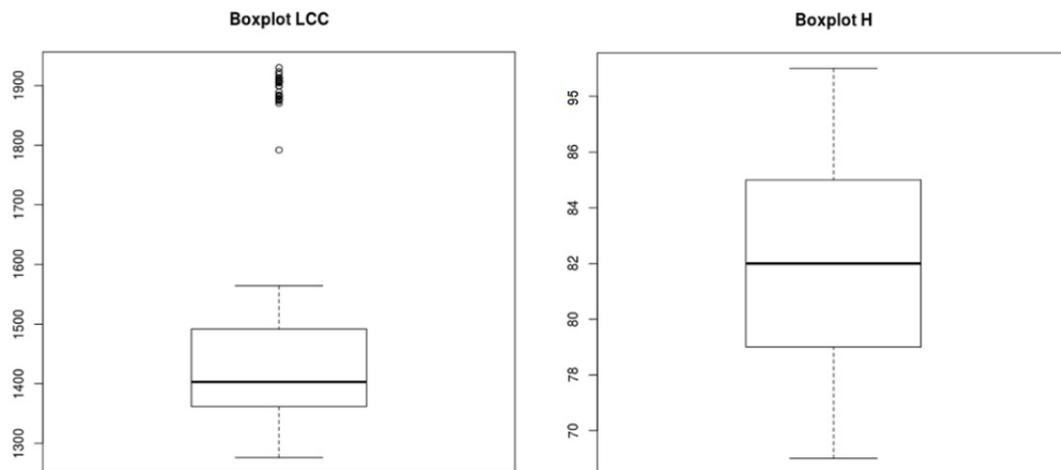


Figura 5.12: Experimento 4 - Boxplots LCC e H (BET)

A Figura 5.13 mostra os gráficos do boxplot das métricas LCC e H para a ALP MOM. A métrica LCC apresenta um intervalo médio entre 750 e 950. A métrica H, por sua vez, exibe intervalo médio entre 27 e 32. Uma vez que a mediana não é equidistante das

extremidades dos boxplots, entende-se que os dados são assimétricos. Ambos os gráficos não revelam *outliers* e os valores estão dispersos dentro dos valores máximos e mínimos. O boxplot de LCC apresenta a mediana próxima do primeiro quartil, ou seja, a maioria dos resultados possuem menos entrelaçamento de características.

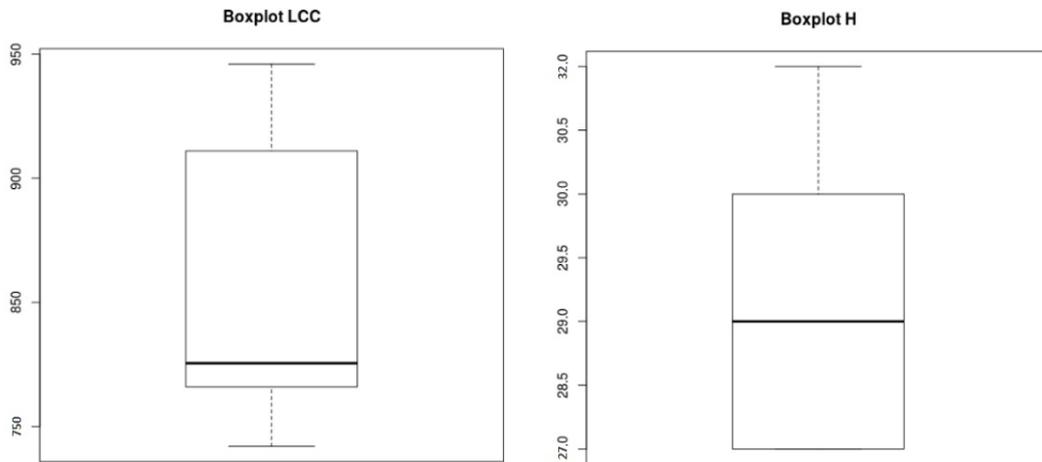


Figura 5.13: Experimento 4 - Boxplots LCC e H (MOM)

Discussões

Considerando a ALP BET e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (LCC;H), é (1913;76), e a solução original de *fitness* igual a (100;88), pode-se observar que houve uma otimização no *outlier* em relação à solução original, pois o número médio de relacionamentos internos entre as classes e interfaces de um componente diminuiu. No entanto, o valor de LCC foi severamente comprometido, pois no *outlier* existem mais elementos (classes, componentes e interfaces) para implementar uma determinada característica, sendo possível observar que no *outlier* foram criadas várias classes e componentes.

É possível observar a correlação negativa entre as funções, pois enquanto o valor de LCC diminui, o de H aumenta. Esse comportamento leva em consideração a comparação entre o *outlier* e a original.

Dessa forma, há evidências para refutar H_{lcc-h0} e aceitar H_{lcc-h1} . Com isso, entende-se que houve correlação negativa significativa muito forte entre as métricas LCC e H.

5.2.5 Experimento 5

No Experimento 5 é analisada a correlação entre as funções ACOMP e TAM, propostas neste trabalho. As variáveis independentes são as ALPs utilizadas, enquanto as funções são as variáveis dependentes.

Hipótese

A hipótese nula é hipótese H_0 , na qual não existe uma correlação significativa entre as funções, e H_1 , em que existe correlação significativa entre as funções ACOMP e TAM.

- h_{h-tam0} : Não existe uma correlação significativa entre H e TAM ;
- H_{h-tam1} : Há correlação significativa entre $ACOMP$ e TAM ;

Resultados

A Tabela 5.16 apresenta os resultados dos testes de normalidade aplicados no Experimento 5. Todos os valores demonstram resultados não normais, portanto, foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.16: Experimento 5 - Teste de Normalidade - Funções ACOMP x TAM

PLA	Funções de Avaliação			
	ACOMP		TAM	
	p-value	Distribuição	p-value	Distribuição
AGM	0.0000000000000000022	Não normal	0.0000000000001237	Não normal
BANK	0.000000000006476	Não normal	0.0000000000494	Não normal
BET	0.0000001119	Não normal	0.000006938	Não normal
MOM	0.0000000003765	Não normal	0.00000000000000022	Não normal

Os resultados do teste de correlação aplicado para o Experimento 5 estão apresentados na Tabela 5.17. Para as quatro ALPs, o teste retornou $p - value < 0.05$ e os valores do nível de correlação negativa significativamente negativa forte para AGM, BANK, BET, assim como a ALP AGM apresentou correlação negativa muito forte.

Tabela 5.17: Experimento 5 - Teste de Correlação - Funções ACOMP x TAM

ALP	Teste Aplicado	Resultados		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.000000000000000022	-0.8349653	Existe Correlação Forte Negativa
BANK	Spearman	0.000000000000000022	-1	Existe Correlação Forte Negativa
BET	Spearman	0.000000000000000022	-0.8239566	Existe Correlação Forte Negativa
MOM	Spearman	0.000000000000000022	-0.9721461	Existe Correlação Muito Forte Negativa

A Figura 5.14 apresenta os boxplots das funções ACOMP e TAM com a ALP AGM. A função ACOMP mostrou uma gama de valores de dados pouco diferentes, com valores divergindo entre um mínimo de 1.0 e um máximo de 3.0, com a mediana entre 1 e 1.5. A mediana não é equidistante dos extremos do boxplot, portanto, pode-se concluir que os dados são assimétricos. Como mostra o gráfico, ocorreu a presença de *outliers*. A função TAM exibe uma variação pouco maior nos valores mínimos e máximos de 30 e 90 e mediana perto de 50. Devido à ocorrência de *outliers* na função ACOMP, foi feito um tratamento destes, conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresenta *outliers*.

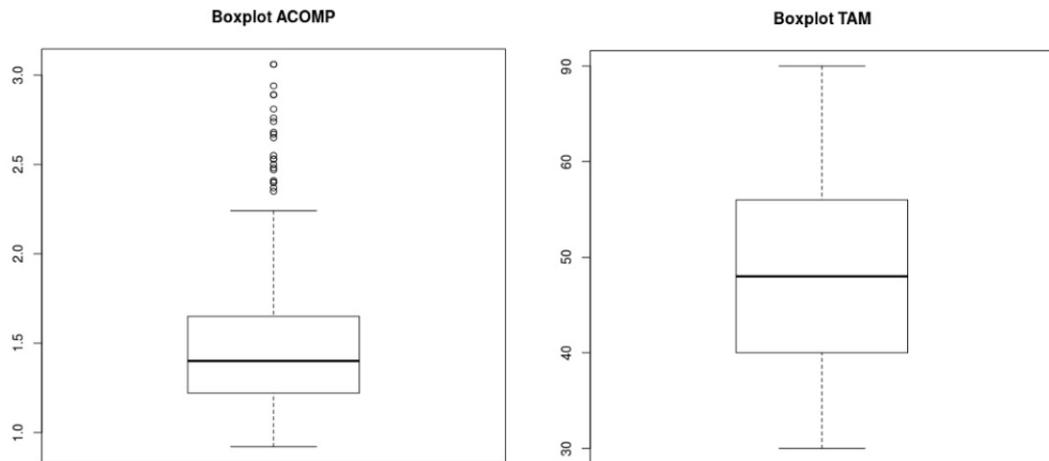


Figura 5.14: Experimento 5 - Boxplots ACOMP e TAM (AGM)

Os valores das funções ACOMP e TAM para a ALP BANK são mostrados na Figura 5.15. A função ACOMP apresentou um intervalo pequeno entre 20 e 40, sem a presença de *outliers*. A função TAM apresentou um intervalo pequeno entre 1.00 e 1.30. O gráfico da função ACOMP é equidistante das medianas da extremidade do boxplot, sendo assim, o resultado da função ACOMP é simétrico. O gráfico da função TAM não apresenta valores equidistantes, portanto, os dados são assimétricos.

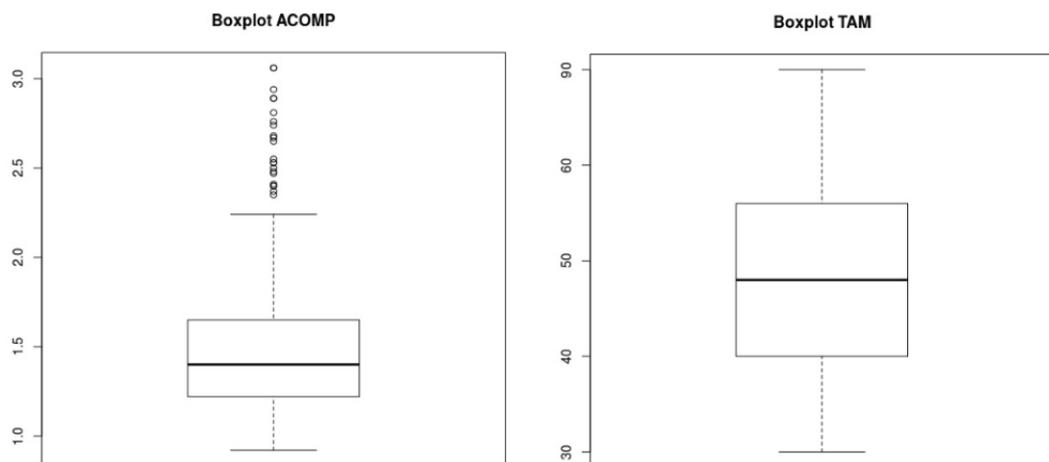


Figura 5.15: Experimento 5 - Boxplots ACOMP e TAM (BANK)

Os valores das funções ACOMP e TAM para a ALP BET são mostrados na Figura 5.16. A função ACOMP apresentou um intervalo médio entre 200 e 320, com a presença de *outliers*. A função TAM, por sua vez, exibiu um intervalo menor ainda, entre 2.0 e 5.5. Ambos os gráficos não são equidistantes das medianas da extremidade do boxplot, sendo assim, o resultado de ambos é assimétrico. Devido à ocorrência de *outliers* na função

ACOMP, foi feito um tratamento destes, conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresenta *outliers*.

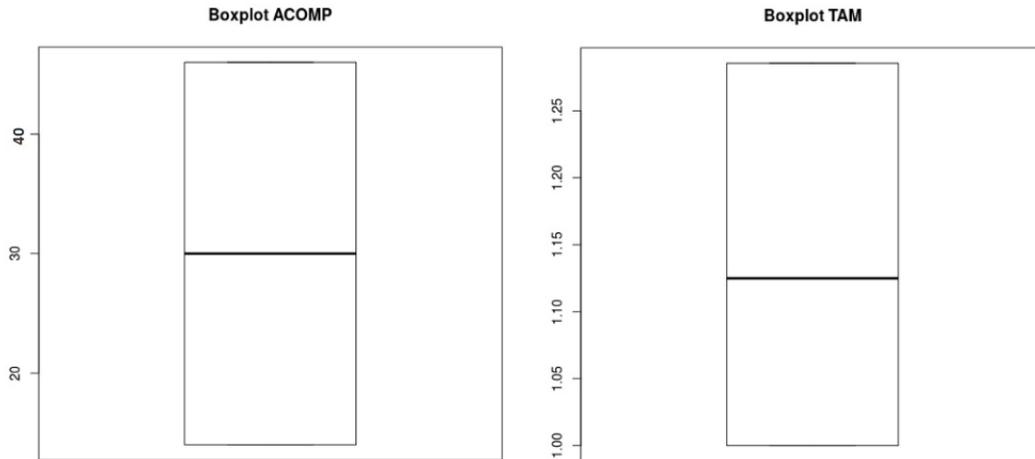


Figura 5.16: Experimento 5 - Boxplots ACOMP e TAM (BET)

A Figura 5.17 mostra os gráficos do boxplot das funções ACOMP e TAM para a ALP MOM. A função ACOMP apresenta um intervalo entre 60 e 120. Já a função TAM aponta valores baixos, entre 2.0 e 4.0. Uma vez que a mediana não é equidistante das extremidades dos boxplots, entende-se que os dados são assimétricos. As funções ACOMP e TAM apresentaram *outliers* para as ALPS AGM e BET, indicando que existe uma alta quantidade de acoplamento de componentes.

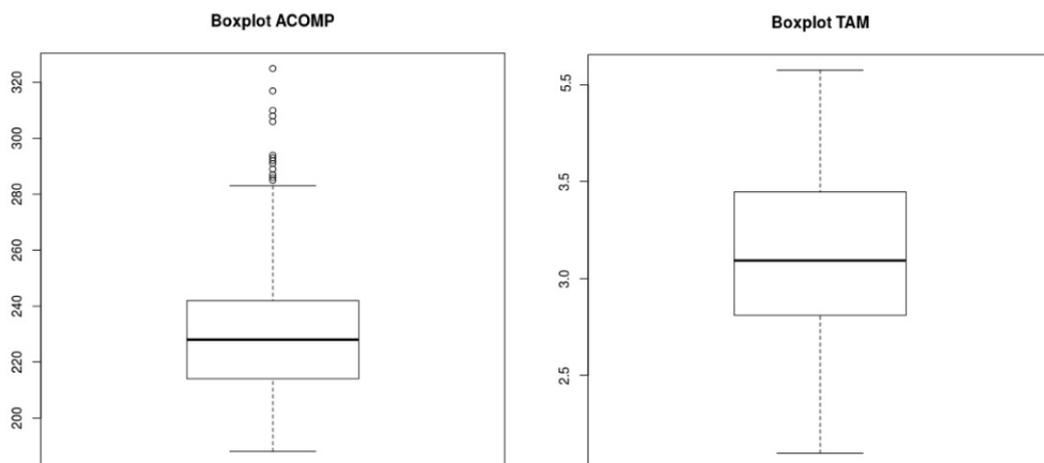


Figura 5.17: Experimento 5 - Boxplots ACOMP e TAM (MOM)

Discussões

Considerando a ALP AGM e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (ACOMP;TAM), é (2.50;70) e a solução original de *fitness* igual a

(3.92;59), pode-se observar que o número de dependências de componentes tanto de entrada como de saída é menor na solução original do que no *outlier*. Isso é percebido no valor do *fitness* referente a ACOMP. Por outro lado, pode-se observar o aumento do número médio de operações por interface (TAM) no *outlier*, em comparação com a solução original.

É possível observar a correlação negativa entre as funções, pois enquanto o valor de ACOMP aumenta, o de TAM diminui. Esse comportamento leva em consideração a comparação entre os *outliers* e a original.

Dessa forma, há evidências para refutar $H_{acomptam0}$ e aceitar $H_{acomptam1}$. Assim, entende-se que houve correlação negativa significativa forte entre as métricas ACOMP e TAM.

5.2.6 Experimento 6

No Experimento 6 é analisada a correlação entre as funções ACLASS e ACOMP, propostas neste trabalho. As variáveis independentes são as ALPs utilizadas, enquanto as funções são as variáveis dependentes.

Hipótese

A hipótese nula é hipótese H_0 , na qual não existe uma correlação significativa entre as métricas, e H_1 , em que existe correlação significativa entre as funções ACLASS e ACOMP.

- $H_{aclass-acomp0}$: Não existe uma correlação significativa entre *ACLASS* e *ACOMP*;
- $H_{aclass-acomp1}$: Há correlação significativa entre *ACLASS* e *ACOMP*.

Resultados

A Tabela 5.18 apresenta os resultados dos testes de normalidade aplicados no Experimento 6. Todos os valores demonstram resultados não normais, portanto, foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.18: Experimento 6 - Teste de Normalidade - Funções ACLASS x ACOMP

PLA	Funções de Avaliação			
	ACLASS		ACOMP	
	p-value	Distribuição	p-value	Distribuição
AGM	0.3178	Não normal	0.000001871	Não normal
BET	0.0000000005365	Não normal	0.000000000000000022	Não normal
MOM	0.0003843	Não normal	0.0000000007448	Não normal

Os resultados do teste de correlação aplicado para o Experimento 6 estão apresentados na Tabela 5.19. Para as três ALPs, o teste retornou $p\text{-value} < 0.05$, indicando correlação positiva moderada para as ALPs BET e MOM e correlação forte para a ALP AGM.

Tabela 5.19: Experimento 6 - Teste de Correlação - Funções ACLASS x ACOMP

ALP	Teste Aplicado	Resultados		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.00002159	0.77899	Existe Correlação Forte Positiva
BET	Spearman	0.000000000000000022	0.53086	Existe Correlação Moderada Positiva
MOM	Spearman	0.000000001047	0.41175	Existe Correlação Moderada Positiva

A Figura 5.18 apresenta os boxplots das funções ACLASS e ACOMP com a ALP AGM. A função ACLASS mostrou uma gama de valores de dados pouco diferentes, com valores variando entre um mínimo de 25 e um máximo de 40. A mediana exibe valor próximo de 33 e não é equidistante dos extremos do boxplot, portanto pode-se concluir que os dados são assimétricos. A função ACOMP revela, por sua vez, uma variação também pequena com valores mínimos e máximos entre 7 e 10, respectivamente, e mediana com valor 9. Como mostra o gráfico ocorreu *outliers* indicando a assimetria dos dados. Devido a essa ocorrência de *outliers*, foi feito um tratamento, do mesmo conforme descrito na Seção 3.2.2 na função ACOMP. No entanto, mesmo com o tratamento, este boxplot ainda apresenta ocorrência de *outliers*.

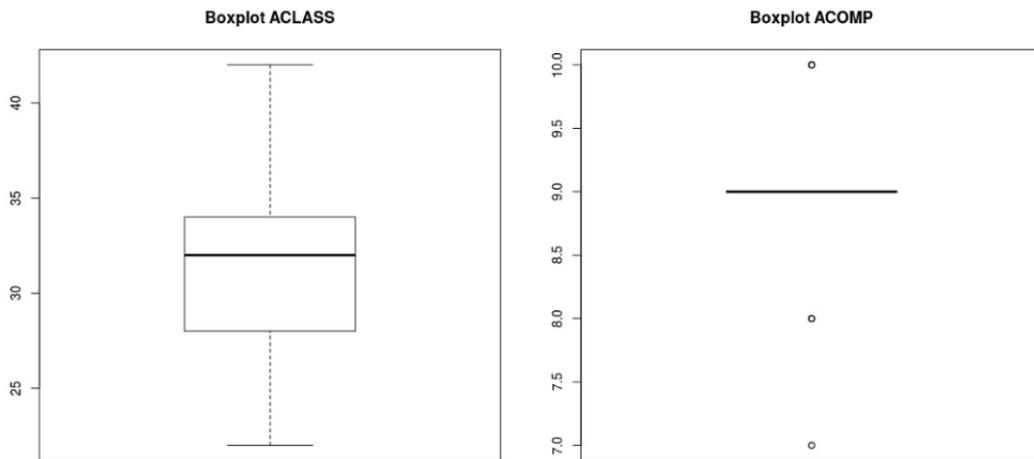


Figura 5.18: Experimento 6 - Boxplots ACLASS e ACOMP (AGM)

Os boxplots referentes às funções ACLASS e ACOMP para a ALP BET são apresentados na Figura 5.19. A função ACLASS exibe valores entre 170 e 205 com mediana próxima de 200, além da presença de *outliers*, demonstrando assimetria dos dados. A função ACOMP, por sua vez, apresentou um intervalo entre 110 e 220 e mediana perto de 120, com a presença de *outliers*, também demonstrando assimetria dos dados. Os gráficos apresentam *outliers* com valores superiores a 75% do número total de soluções. Devido à ocorrência de *outliers*, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, estes boxplots ainda apresentam *outliers*.

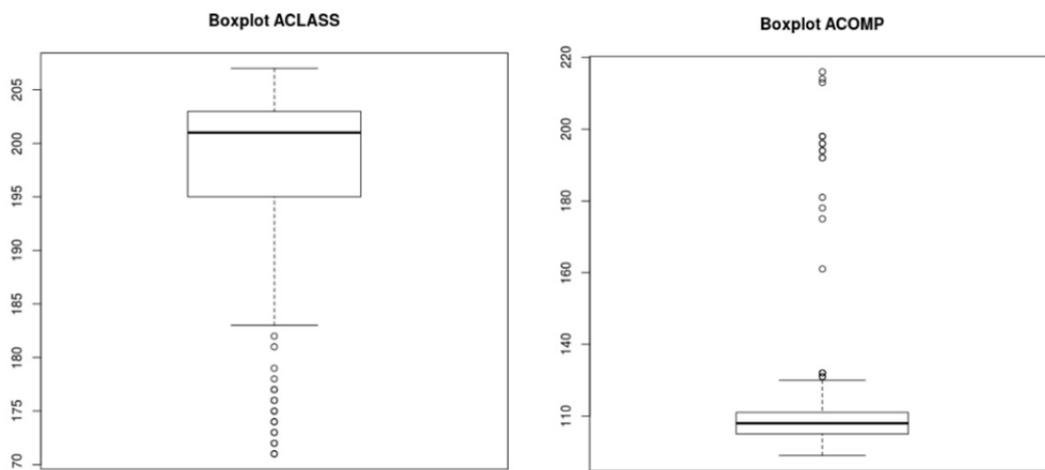


Figura 5.19: Experimento 6 - Boxplots ACLASS e ACOMP (BET)

A Figura 5.20 exibe os gráficos de boxplot das funções ACLASS e ACOMP para a ALP MOM. A função ACLASS apresenta um intervalo entre 25 e 40 com mediana próxima de 36. A função ACOMP demonstra valores entre 6 e 10 e mediana próxima

de 10. Em ambos os boxplots ocorreram *outliers*. Devido à ocorrência destes, foi feito um tratamento conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, estes boxplots ainda apresentam *outliers*. A função ACOMP revela uma concentração de resultados, demonstrando que não foi possível otimizar as ALPs, pois a ALP original é a que apresenta menos acoplamento de componentes.

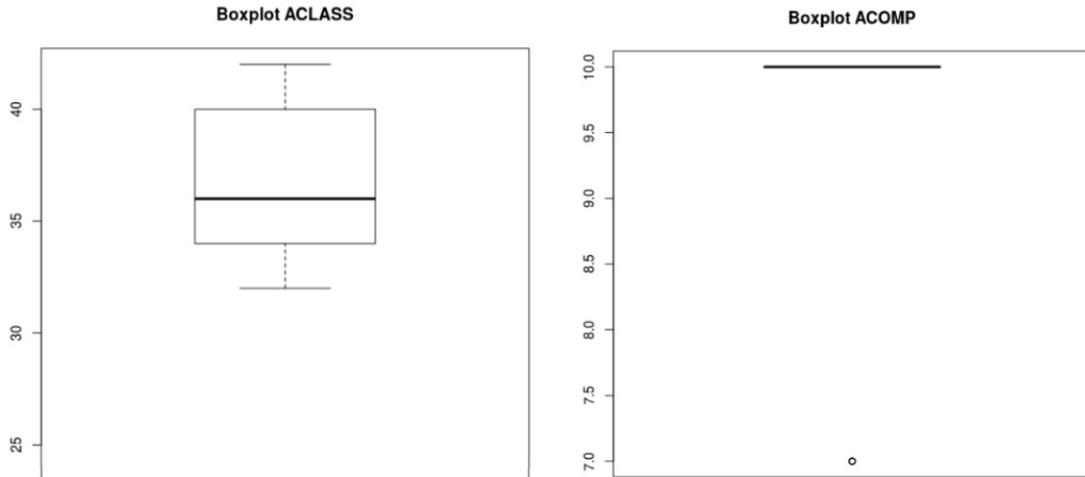


Figura 5.20: Experimento 6 - Boxplots ACLASS e ACOMP (MOM)

Discussões

Consideramos a ALP AGM e tomamos como base para a discussão um *outlier* cujo *fitness*, no formato (AClass;ACOMP), é (40;10) e a solução original de *fitness* igual a (30;3.92). Dado esses valores, pode-se observar que o *outlier* apresenta um maior valor de acoplamento de componentes em relação à solução original. Essa diferença é observada nos valores referentes à ACOMP. Do mesmo modo, é observado um maior valor para o acoplamento das classes do *outlier*, em comparação com a solução original.

Esses valores se justificam, pois o *outlier* mostrou-se dominado pela solução original, considerando que, para ambas as funções, os valores aumentaram. Dados os valores do *outlier*, é possível que este seja uma solução de projeto pior do que a solução original. Apesar dessa situação, observa-se que as funções apresentam valores diretamente proporcionais.

É possível observar a correlação positiva entre as funções, pois quando o valor de ACLASS aumenta, o valor de ACOMP também aumenta. Esse comportamento foi observado considerando o *outlier* e a original.

Nesse sentido, há evidências para refutar $H_{aclass-acomp0}$ e aceitar $H_{aclass-acomp1}$. Logo, entende-se que houve correlação positiva significativa forte entre as métricas ACLASS e ACOMP.

5.2.7 Experimento 7

No Experimento 7 é analisada a correlação entre as funções ACOMP e H. As variáveis independentes são as ALPs utilizadas, enquanto as métricas são as variáveis dependentes.

Hipótese

A hipótese nula é a hipótese H_0 , na qual não existe uma correlação significativa entre as métricas, e H_1 , em que existe correlação significativa entre a função ACOMP e a função H.

- $ACOMP_{acomp-h_0}$: Não existe uma correlação significativa entre $ACOMP$ e H ;
- $ACOMP_{acomp-h_1}$: Há correlação significativa entre $ACOMP$ e H .

Resultados

A Tabela 5.20 apresenta os resultados dos testes de normalidade aplicados no Experimento 7. Todos os resultados apresentam resultados não normais, portanto, foi aplicado o teste de normalidade de Shapiro Wilk.

Tabela 5.20: Experimento 7 - Teste de Normalidade - Função ACOMP x H

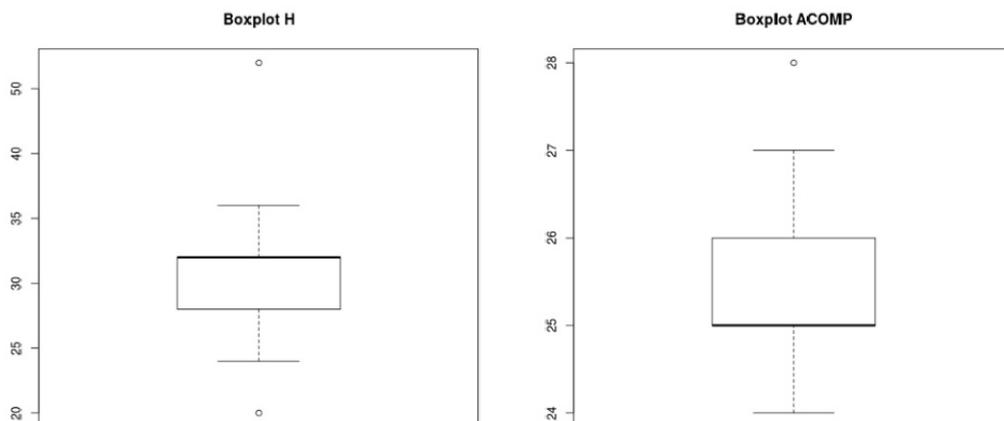
PLA	Funções de Avaliação			
	ACOMP		H	
	p-value	Distribuição	p-value	Distribuição
AGM	0.0000006398	Não normal	0.0000001443	Não normal
BET	0.0006777	Não normal	0.0000005967	Não normal
MOM	0.00000003341	Não normal	0.00000008501	Não normal

Os resultados do teste de correlação aplicado no Experimento 7 estão apresentados na Tabela 5.21. Para as três ALPs, o teste retornou $p - value < 0.05$, indicando correlação moderada positiva.

Tabela 5.21: Experimento 7 - Teste de Correlação - Funções ACOMP x H

ALP	Teste Aplicado	Resultados		
		p-value	Nível de Correlação	Conclusão
AGM	Spearman	0.002089	0.502644	Existe Correlação Moderada Positiva
BET	Spearman	0.0000000000381	0.5233264	Existe Correlação Moderada Positiva
MOM	Spearman	0.0007579	0.5648047	Existe Correlação Moderada Positiva

A Figura 5.21 apresenta os boxplots das funções ACOMP e H da ALP AGM. A função ACOMP mostrou uma gama de valores pouco diferentes, com valores divergindo entre um mínimo de 24 e um máximo de 28 e mediana 25. A mediana não é equidistante dos extremos do boxplot, portanto, pode-se concluir que os dados são assimétricos. A função H apresenta uma variação também pequena com valores mínimos e máximos entre 20 e 50 e mediana próxima de 33 com dados assimétricos. Como mostram os gráficos, ocorreram *outliers*. Devido à ocorrência de destes em ambas as funções, foi feito um tratamento dos mesmos conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, estes boxplots ainda apresentam *outliers*.

**Figura 5.21:** Experimento 7 - Boxplots ACOMP e H (AGM)

A Figura 5.22 mostra os valores das função ACOMP e H para a ALP BET. A função ACOMP apresentou um valor médio entre 75 e 82 e mediana com valor 78. A métrica H, por sua vez, demonstrou um intervalo maior entre 210 e 280 e mediana próxima a 225, revelando a assimetria dos dados para ambas as funções.

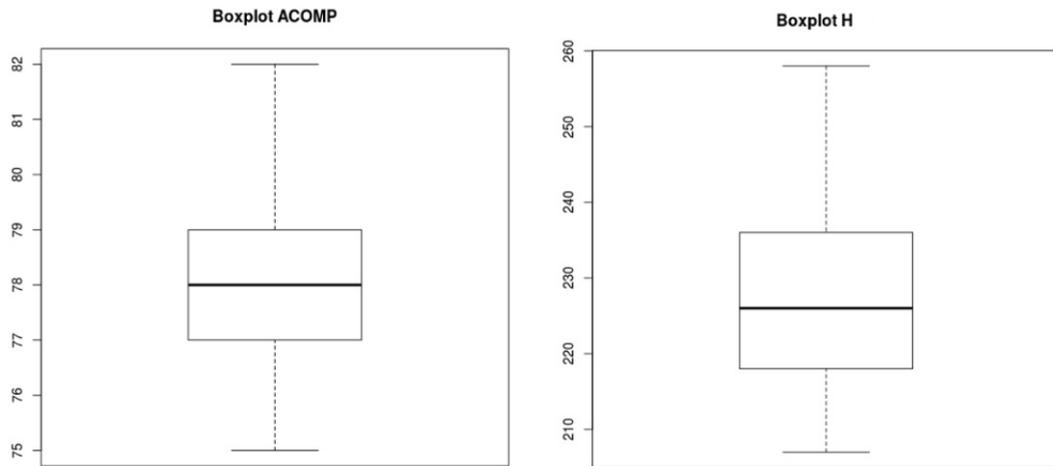


Figura 5.22: Experimento 7 - Boxplots ACOMP E H (BET)

A Figura 5.23 mostra os gráficos do boxplot das funções ACOMP e H para a ALP MOM. A função H apresenta um intervalo entre 46 e 52 e mediana 49. Já a função ACOMP apresenta valores entre 27 e 31 e mediana 28, porém, com a presença de *outliers*, revelando a assimetria dos dados em ambas as funções. Devido à ocorrência de *outliers* na função ACOMP, foi feito um tratamento do mesmo conforme descrito na Seção 3.2.2. No entanto, mesmo com o tratamento, este boxplot ainda apresentam *outliers*.

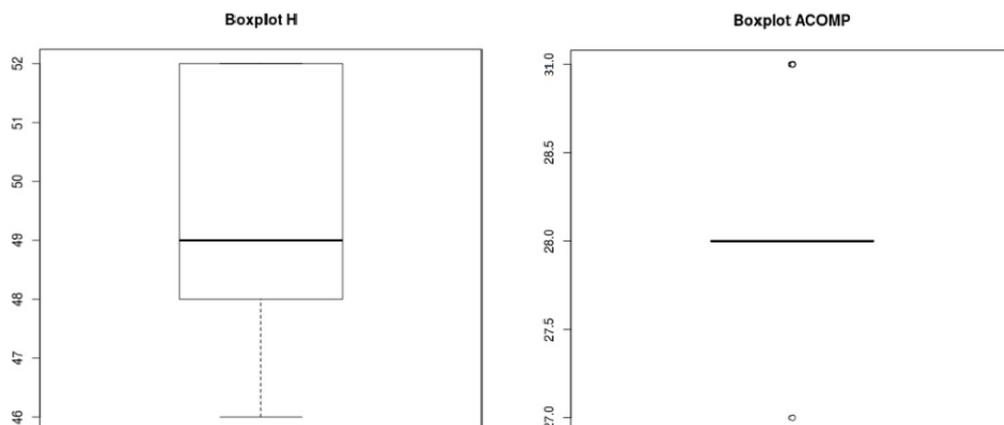


Figura 5.23: Experimento 7 - Boxplots ACOMP e H (MOM)

Discussões

Considerando a ALP AGM e tomando como base para a discussão um *outlier* cujo *fitness*, no formato (ACOMP;H), é (28;20) e a solução original de *fitness* igual a (3.98;7), pode-se observar que o número de dependências de componentes tanto de entrada como de saída é menor na solução original do que no *outlier*, conforme observado no valor *fitness* do *outlier* com relação à função ACOMP. Do mesmo modo, o número de relacionamentos internos entre classes e interfaces de um componente específico é menor na solução original em relação ao *outlier*.

É possível observar a correlação positiva entre as funções, pois quando o valor de ACOMP diminui, o valor de H também diminui. Esse comportamento foi observado considerando o *outlier* e a original.

Dessa forma, existem evidências para refutar $H_{acomp-h0}$ e aceitar $H_{acomp-h1}$, pois houve correlação positiva moderada entre as funções ACOMP e H.

5.2.8 Análise dos Resultados dos Experimentos de Correlação

Teste de Efeito

O teste de efeito de Cliff's (Vaidyanathan, 1993) foi aplicado com a intenção de analisar o efeito das novas funções de avaliação em relação às funções de avaliação FM(pla) e CM(pla). As Tabelas 5.22, 5.23 e 5.24 apresentam o teste de efeito de Cliff's aplicado com as funções DC(pla), EC(pla) e LCC(pla) com a função FM(pla). As Tabelas 5.25, 5.26 e 5.27 apresentam o teste de efeito de Cliff's aplicado com as funções ACLASS(pla), ACOMP(pla), TAM(pla) e H(pla).

Alguns resultados exibem valores negativos, porém para o teste de efeito de Cliff's isso não interfere no resultado (efeito). Isso porque em uma comparação de conjuntos A e B, se o valor de A tem o efeito de tamanho maior que B, o valor é positivo. Se B for maior que A o valor vai ser negativo, sendo assim, se inverter a ordem do teste de B para A, o valor do efeito fica positivo, porque B é o maior.

A Tabela 5.22 apresenta os resultados do teste de efeito entre a função FM(pla) e as funções DC(pla) e EC(pla) para o Experimento 1. Para os resultados de EC(pla) com as ALPs AGM e MOM, obteve-se resultado desprezível, pois a diferença de valores é muito grande entre o tratamento e controle. Na função DC(pla) para a ALP AGM o teste retornou um número alto em relação à escala do tamanho do efeito, sendo assim, a função FM(pla) obteve mais efeito que DC(pla), pois para ambas as ALPs os resultados foram muito altos.

A Tabela 5.23 apresenta os resultados do teste de efeito entre a função FM(pla) e as funções DC(pla) e LCC(pla) para o Experimento 2. Para os resultados de LCC(pla)

Tabela 5.22: Experimento 1 - Teste de Efeito de Cliff's FM(pla)

Experimento 1		
ALP	Função	Resultado do Teste
AGM	EC(pla)	0 (Desprezível)
	DC(pla)	0.8381643 (Muito Grande)
MOM	EC(pla)	0 (Desprezível)
	DC(pla)	1 (Muito Grande)

com as ALPs AGM e MOM, obteve-se resultado desprezível, pois a diferença de valores é muito grande entre o tratamento e controle. Na função DC(pla) para a ALP AGM o teste retornou um número alto em relação à escala do tamanho do efeito, sendo assim a função FM(pla) obteve mais efeito que DC(pla), pois para ambas as ALPs os resultados foram muito altos.

Tabela 5.23: Experimento 2 - Teste de Efeito de Cliff's FM(pla)

Experimento 2		
ALP	Função	Resultado do Teste
AGM	DC(pla)	0.8381643 (Muito Grande)
	LCC(pla)	0 (Desprezível)
MOM	DC(pla)	1 (Muito Grande)
	LCC(pla)	0 (Desprezível)

A Tabela 5.24 exibe os resultados do teste de efeito entre a função FM(pla) e as funções EC(pla) e LCC(pla) para o Experimento 3. Para todas as funções e ALPs, obteve-se resultado muito alto em relação à escala do tamanho do efeito. No entanto a função EC(pla) obteve resultado positivo, mostrando que FM(pla) obteve mais efeito que EC(pla) para a ALP MOM(pla), no entanto, EC(pla) e LCC(pla) obtiveram mais efeito que FM(pla), pois para a ALP AGM(pla) ambos os resultados resultados foram negativos.

Tabela 5.24: Experimento 3 - Teste de Efeito de Cliff's FM(pla)

Experimento 3		
ALP	Função	Resultado do Teste
AGM	EC(pla)	-1 (Muito Grande)
	LCC(pla)	-1 (Muito Grande)
MOM	EC(pla)	1 (Muito Grande)
	LCC(pla)	-1 (Muito Grande)

A Tabela 5.25 apresenta os resultados do teste de efeito entre a função FM(pla) e as funções LCC(pla) e H(pla) para o Experimento 4. Para os resultados de LCC(pla) com as

ALPs AGM e MOM, obteve-se resultado muito grande com os valores 1 para a ALP AGM e 0.9837234 para a ALP MOM, sendo assim, pode-se dizer que FM(pla) obteve mais efeito que LCC(PLA). Na função H(pla), para ambas as ALPs AGM e MOM, o teste retornou um número negativo alto em relação à escala do tamanho do efeito, de modo que sendo assim pode-se afirmar que H(pla) tem mais efeito que CM(pla).

Tabela 5.25: Experimento 4 - Teste de Efeito Cliff's FM(pla)

Experimento 4		
ALP	Função	Resultado Teste
AGM	LCC(pla)	1 (Muito Grande)
	H(pla)	-1 (Muito Grande)
MOM	LCC(pla)	0.9837234 (Muito Grande)
	H(pla)	-1 (Muito Grande)

A Tabela 5.26 apresenta os resultados do teste de efeito entre a função CM(pla) e as funções ACOMP(pla) e TAM(pla) para o Experimento 5. Os resultados mostram que a ALP AGM(pla), a função ACOMP(pla) obteve mais efeito que CM(pla), e CM(pla) obteve mais efeito que TAM(pla). Na ALP MOM(pla), a função CM(pla) obteve mais efeito que ACOMP(pla) e TAM(pla) obteve mais efeito que CM(pla).

Tabela 5.26: Experimento 5 - Teste de Efeito de Cliff's CM(pla)

Experimento 5		
ALP	Função	Resultado do Teste
AGM	ACOMP(pla)	- 1 (Muito Grande)
	TAM(pla)	1 (Muito Grande)
MOM	ACOMP(pla)	1 (Muito Grande)
	TAM(pla)	-1 (Muito Grande)

A Tabela 5.27 exhibe os resultados do teste de efeito entre a função CM(pla) e as funções ACLASS(pla) e ACOMP(pla) para o Experimento 6. Os resultados mostram que CM(pla) obteve mais efeito que ACLASS(pla), pois ambos os valores são positivos e muito altos. A função ACOMP(pla) obteve mais efeito que CM(pla), pois os resultados são negativos e muito altos.

Tabela 5.27: Experimento 6 - Teste de Efeito de Cliff's CM(pla)

Experimento 6		
ALP	Função	Resultado do Teste
AGM	AClass(pla)	1 (Muito Grande)
	ACOMP(pla)	-1 (Muito Grande)
MOM	AClass(pla)	0.6428571 (Muito Grande)
	ACOMP(pla)	-1 (Muito Grande)

A Tabela 5.28 apresenta os resultados do teste de efeito entre a função CM(pla) e as funções ACOMP(pla) e H(pla) para o Experimento 7. Os resultados mostram que ACOMP(pla) obteve mais efeito que CM(pla), pois seus valores são negativos e muito altos. A função CM(pla), obteve mais efeito que H(pla) com a ALP MOM(pla), e H(pla) obteve mais efeito que CM(pla) com a ALP AGM, pois um resultado foi positivo e outro negativo.

Tabela 5.28: Experimento 7 - Teste de Efeito de Cliff's CM(pla)

Experimento 7		
ALP	Função	Resultado do Teste
AGM	ACOMP(pla)	-1 (Muito Grande)
	H(pla)	0.9807692 (Muito Grande)
MOM	ACOMP(pla)	-1 (Muito Grande)
	H(pla)	-0.7606838 (Muito Grande)

Os experimentos mostraram que apesar da função de avaliação formada pelas funções dirigidas à características (FM) não ter sido tão sensível quanto as funções de avaliação DC, EC e LCC, a função de avaliação composta pelas métricas convencionais (CM) foram predominantemente melhores que as funções de avaliação que contém essas métricas isoladas (ACOMP, ACLASS, TAM e H). Uma comparação quantitativa mostra que a função CM teve mais efeito em 5 casos enquanto as as funções isoladas das métricas convencionais tiveram mais efeito em nove situações. Esses resultados mostram que ter separado as métricas convencionais em novas funções de avaliação, foi melhor, pois as novas funções apresentam mais efeito no resultados da otimização, se comparado com as funções CM e FM.

Análise da porcentagem

A Tabela 5.29 apresenta os melhores e os piores valores de FM(pla)e CM(pla), assim como os percentuais de melhoria da função FM(pla) em relação ao *fitness* original. As informações dessa tabela foram obtidas em um estudo realizado em (Colanzi et al., 2014).

Como as funções CM e FM são conflitantes, a coluna que mostra o fitness da solução de melhor FM é, por consequência, a solução com o pior CM e vice-versa.

Tabela 5.29: Porcentagem de melhoria NSGA-II (Colanzi et al., 2014)

PLA	NSGA-II	
	Melhor FM (FM, CM)	Pior FM (FM, CM)
AGM-1	(181.0, 21.25) 53.70%, 33.21%	(332.0, 19.25) 15.08%, 39.50%
MM-1	(219.0, 26.20) 63.98%, 33.35%	(437.0, 26.08) 28.12%, 33.65%
BET	(476.0, 177.43) 29.37%, -0.33%	(615.0, 153.14) 8.75%, 13.40%

As Tabelas 5.30, 5.31, 5.32, 5.33, 5.34, 5.35 e 5.36 apresentam os resultados referentes ao *fitness* original, o melhor e o pior resultado de cada conjunto de valores para cada uma das ALPs utilizadas nos experimentos. Estas tabelas expõem os percentuais de melhoria em relação ao *fitness* original, assim como a Tabela 5.29, com o objetivo de comparar os percentuais de melhoria em relação a função FM(pla) com cada uma das novas ALPs propostas neste trabalho.

A Tabela 5.30 exibe os valores das funções EC(pla) e DC(pla). As novas funções apresentam uma porcentagem menor em relação a FM(pla) da Tabela 5.29, por exemplo, comparando os valores de melhor EC com os de melhor FM, dois resultados revelam piores percentuais para EC (50% e 53.77%). Se comparar os resultados de melhor DC com os melhores de FM(pla), DC apresenta menores percentuais (43.48%, 27% e 36%). Comparando os piores resultados, FM(pla) teve maiores percentuais do que EC e DC. Estes resultados mostram que a função FM(pla) apresenta melhor percentual de melhoria em relação a EC(pla) e DC(pla) mas também maiores percentuais no pior valor da função.

Tabela 5.30: Porcentagem de melhoria Experimento 1 - DCxEC

Experimento 1					
PLA	Fitness Original (EC, DC)	Melhor EC (EC, DC)	Pior EC (EC, DC)	Melhor DC (EC, DC)	Pior DC (EC, DC)
AGM	124, 430	62.0, 247.0 50%, 42.55%	96.0, 319.0 22.58%, 26.51%	62.0, 243.0 50%, 43.48%	82.0, 369.0 33.87%, 14.18%
MM	225, 680	104.0, 556.0 53.77%, 18.23%	161.0, 505.0 28.44%, 25.75%	133.0, 496.0 40.88%, 27%	121.0, 562.0 46.22%, 17.35%
BET	200, 688	137.0, 547.0 31.5%, 20.49%	235.0, 483.0 -17.5%, 29.79%	220.0, 435.0 -10.0%, 36%	173.0, 611.0 13.5%, 11.19%

A Tabela 5.31 exibe os valores das funções DC(pla) e LCC(pla). As novas funções apresentam uma porcentagem menores em relação a FM(pla) da Tabela 5.29, por exemplo, comparando os valores de melhor DC com os de melhor FM, DC possui dois resultados com menores percentuais (27.44% e 6.47%) em relação à FM(pla). A função LCC(pla) apresentou percentuais negativos e zerados, portanto menores que FM(pla). Comparando os percentuais de piora, é possível observar que FM(pla) obtiveram percentuais maiores que as funções DC(pla) e LCC(pla), pois apresentam percentuais um pouco maiores, por exemplo 9.30% para DC(pla) e 8.75% para FM(pla). Estes resultados corroboram a análise de correlação fraca ou moderada entre as novas funções de avaliação.

Tabela 5.31: Porcentagem de melhoria Experimento 2 - DCxLCC

Experimento 2					
PLA	Fitness Original (DC, LCC)	Melhor DC (DC, LCC)	Pior DC (DC, LCC)	Melhor LCC (DC, LCC)	Pior LCC (DC, LCC)
AGM	430, 26	312.0, 35.0 27.44%, -34%	460.0, 25.0 -6.97, 3.84%	451.0, 25.0 -4.88%, 3,84%	316.0, 36.0 25.51%, -38%
MM	680, 28	636.0, 31.0 6,47%, -10%	680.0, 28.0 0%, 0%	680.0, 28.0 0%, 0%	636.0, 31.0 6,47%, -10.71%
BET	688, 100	442.0, 220.0 35.75%, -120%	624.0, 172.0 9.30%, -72%	513.0, 137.0 25.43%, -37%	477.0, 241.0 30.66%, -141%

A Tabela 5.32 apresenta os valores das funções EC(pla) e LCC(pla). As novas funções exibem uma porcentagem menor em relação à função FM(pla) da Tabela 5.29, em que todos os valores de EC são menores em relação à função FM(pla) (50%, 53.77% e 8.5%) comparando o melhor EC com o melhor FM. Comparando os percentuais de piora, dois valores de EC (1.61% e 26.22%) são menores que FM(pla). Estes resultados mostram que a função FM(pla) obtive percentual melhor que as novas funções de avaliação.

Tabela 5.32: Porcentagem de melhoria Experimento 3 - ECxLCC

Experimento 3					
PLA	Fitness Original (EC, LCC)	Melhor EC (EC, LCC)	Pior EC (EC, LCC)	Melhor LCC (EC, LCC)	Pior LCC (EC, LCC)
AGM	124, 26	62.0, 41.0 50%, -57.69%	122.0, 25.0 1,61%, 3,84%	122.0, 25.0 1,61%, 3,84%	62.0, 42.0 50%, -61.53%
MM	225, 28	104.0, 33.0 53.77%, -17.85%	166.0, 27.0 26.22%, 3,57%	166.0, 27.0 26.22%, 3,57%	104.0, 36.0 53.77%, -28%
BET	200, 100	183.0, 95.0 8.5%, 5%	253.0, 76.0 -26.5%, 24%	253.0, 76.0 -26.5%, 24%	183.0, 95.0 8.5%, 5%

A Tabela 5.33 exibe os valores das funções LCC(pla) e H(pla). As novas funções apresentam altas porcentagens negativas. Nesse experimento, há percentuais positivos de melhoria de fitness somente para a PLA AGM. Comparando os percentuais de H com os valores de CM(pla), é possível observar que todas as porcentagens (melhor e pior) de H são maiores que as de CM, evidenciando que uma função de avaliação que agrega várias métricas não é tão sensível quanto a função H.

Tabela 5.33: Porcentagem de melhoria Experimento 4 - LCCxH

Experimento 4					
PLA	Fitness Original (LCC, H)	Melhor LCC (LCC, H)	Pior LCC (LCC, H)	Melhor H (LCC, H)	Pior H (LCC, H)
AGM	26, 7	25.0, 753.0 4%,-1.065%	41.0, 571.0 -57%,-805%	35.0, 553.0 -34.61%,-7.800%	25.0, 753.0 4%,-10.657%
MM	28, 7	788.0, 32.0 -2.714%,-357%	918.0, 28.0 -3.178%,-300%	911.0, 28.0 -3.153%,-300%	808.0, 31.0 -2.785%,-342%
BET	100, 48	1281.0, 91.0 -1.181%,-89%	1913.0, 76.0 -1.813%,-58%	1541.0, 77.0 -1.441%,-60%	1281.0, 91.0 -1.181%,-89%

A Tabela 5.34 exibe os valores das funções ACOMP(pla) e TAM(pla). As novas funções apresentam somente um valor (6.89%) menor em relação à CM(pla) da Tabela 5.29. A função TAM(pla) apresenta somente o valor 14.15% menor em relação à CM(pla), e também apresenta valores negativos baixos. Quando se compara os resultados de pior porcentagem, o mesmo se repete, pois somente o valor 10.34% da função TAM(pla) apresenta porcentagem menor. Estes resultados mostram que a função CM(pla) obteve melhores percentuais que ACOMP(pla) e TAM(pla), indicando que as funções com métricas isoladas são mais sensíveis às alterações no projeto de PLA.

Tabela 5.34: Porcentagem de melhoria Experimento 5 - ACOMPxTAM

Experimento 5					
PLA	Fitness Original (ACOMP, TAM)	Melhor ACOMP (ACOMP,TAM)	Pior ACOMP (ACOMP,TAM)	Melhor TAM (ACOMP,TAM)	Pior TAM (ACOMP,TAM)
AGM	1.8, 24	0.91, 66 98%, -1.583%	2.05, 34 96%, -767%	2.4, 34 95%, -767%	1.00, 80 98%, -1.940%
BAN	59.0, 3.92	22.0, 1.28 -1.122%, 94%	38.0, 1.0 -2.011%, 95%	38.0, 1.0 -2.011%, 95%	22.0, 1.28 -1.122%, 94%
MM	58.0, 5.86	54.0, 3.19 6.89%, 45%	98.0, 1.75 -68.96%, 70%	98.0, 1.75 -68.96%, 70%	52.0, 3.83 10.34%, 34%
BET	339.0, 5.4	200.0, 3.81 41%, 29%	293.0, 2.40 13.56%, 55%	291, 2.42 14.15%, 55%	209.0, 4.82 38.34%, 10%

A Tabela 5.35 exibe os valores das funções ACLASS(pla) e ACOMP(pla). Os melhores valores de ACLASS apresentam menores percentuais de melhoria do que CM(pla) (Tabela 5.29). Ao se comparar os valores de melhor ACOMP(pla) com melhor CM percebe-se que ACOMP obteve melhor percentual de melhoria para todas as PLAs, evidenciando também que a função é mais sensível às alterações no projeto do que a função CM.

Tabela 5.35: Porcentagem de melhoria Experimento 6 - ACLASSxACOMP

Experimento 6					
PLA	Fitness Original (AClass, ACOMP)	Melhor ACLASS (AClass, ACOMP)	Pior ACLASS (AClass, ACOMP)	Melhor ACOMP (AClass, ACOMP)	Pior ACOMP (AClass, ACOMP)
AGM	30, 59	26.0, 9.0 13.33%, 84%	42.0, 9.0 -40%, 84%	28.0, 8.0 6.66%, 86%	32.0, 13.0 -6.66%, 77%
MM	14, 58	24.0, 7.0 -71.42%, 87%	42.0, 10.0 -200%, 82%	34.0, 6.0 -142%, 89%	34.0, 10.0 -142%, 82%
BET	122, 339	171.0, 198.0 -45%, 41.59%	207.0, 115.0 -69%, 66%	178.0, 109.0 -45.90%, 67%	184.0, 462.0 -50%, -36%

A Tabela 5.36 exibe os valores das funções ACOMP(pla) e H(pla). As novas funções apresentam valores maiores de porcentagem em relação a CM(pla) da Tabela 5.29, por exemplo, todos os valores de ACOMP(pla) são maiores que CM(pla) e todos os valores de H(pla) são maiores e negativos se comparado com CM(pla). As porcentagens de piores exibem porcentagem maiores e negativos em relação CM(pla), o que evidencia um balanceamento entre as novas funções e a CM(pla).

Tabela 5.36: Porcentagem de melhoria Experimento 7 - ACOMPxH

Experimento 7					
PLA	Fitness Original (ACOMP, H)	Melhor ACOMP (ACOMP, H)	Pior ACOMP (ACOMP, H)	Melhor H (ACOMP, H)	Pior H (ACOMP, H)
AGM	7, 59	24.0, 32.0 59%, -357%	27.0, 28.0 54%, -300%	25.0, 28.0 57%, -300%	25.0, 49.0 57%, -600%
MM	7, 58	29.0, 46.0 50%, -557%	29.0, 50.0 50%, -614%	27.0, 46.0 53%, -557%	28.0, 56.0 51%, -700%
BET	48, 339	75.0, 239.0 77.87%, -397%	82.0, 229.0 75.81%, -337%	80.0, 207.0 76%, -331%	76.0, 291.0 77%, -506%

Os percentuais negativos mostram a necessidade de funções objetivo separadas para ter maior sensibilidade no processo de avaliação tendo em vista que os percentuais de CM e FM são positivos em sua maioria. Isso reflete o impacto dos valores das métricas nas funções de avaliação além de confirmar a necessidade de ter funções de avaliação que agreguem um número menor de métricas ou, ao menos, métricas com as mesmas grandezas.

5.2.9 Ameaças à Validade

Há três ameaças à validade que devem ser considerados nestes experimentos. A primeira está relacionada com o tamanho das PLAs utilizadas, pois as ALPs utilizados são de ambiente acadêmico e possuem complexidade reduzida. O segundo risco encontrado é relativo ao número de ALPs utilizadas, uma vez que apenas quatro PLAs foram utilizadas. Ambos os riscos podem ser mitigados através da realização de novos experimentos, utilizando um maior número de ALPs e, também, ALPs mais complexas. A terceira e última ameaça está relacionada com a aleatoriedade dos resultados encontrados pelos algoritmos de busca, neste caso, o NSGA-II. A fim de resolver tal ameaça, cada experimento foi executado 30 (trinta) vezes, permitindo analisar estatisticamente os resultados (Arcuri e Briand, 2014).

5.3 Considerações Finais

Estes estudos experimentais tiveram como objetivo analisar uma possível correlação entre as funções de avaliação propostas no Capítulo 4, a saber, as funções ACLASS, ACOMP, TAM, DC e EC, e uma variação da função COE, utilizando as métricas LCC e H, separadamente. O objetivo foi analisar a correlação entre estas funções e, assim, uma vez correlacionadas, usar essa informação para apoiar o arquiteto de LPS na seleção dos objetivos a serem otimizados na MOA4PLA.

Os resultados empíricos fornecem evidências de correlação negativa significativa em cinco pares de funções analisados e correlação positiva moderada em dois pares. Os Experimentos 1, 2, 3, 4 e 5 apresentaram correlação negativa significativa, e os Experimento 6 e 7, correlação positiva moderada. Dessa forma, nas funções que exibem correlação negativa, quando um objetivo melhora, o outro piora. Nas funções que revelam correlação positiva, por sua vez, ambas as funções ou melhoram, ou pioram. Logo, é possível afirmar que se um arquiteto precisa otimizar uma LPS com mais de um objetivo, ele deverá escolher as funções de avaliação referentes aos objetivos a serem priorizados. Vale ressaltar que, conforme o estudo exploratório corroborou, o desempenho dos MOEAS fica comprometido na presença de muitos objetivos. De acordo com a recomendação da literatura (Deb e Jain, 2012), para obter o melhor desempenho desses algoritmos, não se deve ultrapassar 4 objetivos.

O Experimento 4 mostrou que LCC e H são negativamente correlacionadas e, por isso, não faz sentido comporem a função COE. Desse modo, propõe-se duas novas funções (uma só com LCC e outra com H). Abaixo, seguem as novas equações proposta. A primeira equação é da função H:

$$H(alp) = \sum_{i=1}^c H \quad (5.1)$$

A segunda função é LCC:

$$LCC(alp) = \sum_{i=1}^c LCC \quad (5.2)$$

A Tabela 5.37 apresenta todas as funções de avaliação da MOA4PLA:

Tabela 5.37: Modelo de Avaliação da MOA4PLA

Hierarquia	Nome da Função	Propósito da Função de Avaliação	Métricas
CM(pla)	CM(pla)	Coesão e Acoplamento	DepIn, DepOut, CDepIn, CDepOut, DepPack, NumOps e H.
	ACOMP(pla)	Acoplamento de Componentes	DepIn e DepOut
	ACCLASS(pla)	Acoplamento de Classes	CDepIn e CDepOut
	TAM(pla)	Tamanho	NumOps
	H(pla)	Coesão Relacional	H
FM(pla)	FM(pla)	Modularização de Características	LCC, CDAC, CDAI, CDAO, CIBC, IIBC e OOBC
	LCC(pla)	Número de Interesses Distintos	LCC
	EC(pla)	Entrelaçamento de Características	CIBC, IIBC e OOBC
	DC(pla)	Difusão de Característica	CDAI, CDAO e CDAC
Ext(pla)	Ext(pla)	Extensibilidade	Extens
Eleg(pla)	Eleg(pla)	Elegância	NAC, EC e ATMT

Dessa maneira, o modelo de avaliação fica composto por 11 funções de avaliação (CM(pla), FM(pla), Ext(pla), Eleg(pla), ACOMP(Pla), ACLASS(pla), TAM(pla), H(pla), LCC(pla), EC(pla) e DC(Pla), conforme a Tabela 5.37.

Uma possível explicação para a correlação negativa entre as funções de avaliação é que o operador de mutação *Feature-driven Mutation* da MOA4PLA cria novos componentes e modulariza características em LPS, dessa maneira, os elementos arquiteturais são reorganizados com a intenção de se obter uma ALP modularizada. Ao se modularizar

uma determinada característica, a difusão de características (*DC*) tende a diminuir, o que implicaria, também, reduzir a falta de coesão baseada em recursos (*LCC*). No entanto, existem elementos arquiteturais de ALPs utilizados neste trabalho que realizam mais de uma característica, o que apresenta dificuldades à modularização completa de todas as características e tem um impacto negativo sobre entrelaçamento de características (*EC*) e na falta de coesão (*LCC*), uma vez que os novos componentes foram adicionados no projeto arquitetural para realizar as características que o operador de mutação tentou modularizar.

Devido ao grande número de soluções geradas no Experimento 03, é possível afirmar que existe um forte conflito entre as funções de avaliação *EC* e *LCC*. Além disso, neste experimento, todas as amostras de dados eram assimétricas. Um estudo qualitativo das soluções geradas ajudaria a compreender em que medida o conflito acontece. A ALP BET parece ser diferente dos outros dois modelos, porque todos os seus resultados são assimétricos. Tal estudo qualitativo contribui também para compreender particularidades deste projeto PLA, que é a maior desta pesquisa experimental.

O fato do operador de mutação *Feature-driven Mutation* da MOA4PLA criar novos componentes pode ter influenciado na correlação positiva entre acoplamento de classes e acoplamento de componentes, pois os novos componentes trazem classes antes de outros componentes e gera novas classes e interfaces.

O teste estatístico apresenta os *outliers* como valores considerados fora do padrão, no entanto, *outliers* não são necessariamente soluções ineficazes, ou seja, podem ser factíveis, pois podem representar soluções em que um objetivo foi muito otimizado, comprometendo gravemente o outro, o que levou a valores extremos. Nesse sentido, será preciso fazer uma análise qualitativa das soluções em trabalhos futuros.

Comparando os testes estatísticos das amostras originais com *outliers*, é possível visualizar uma variação na intensidade da correlação. De qualquer forma, ambas as amostras apontam para uma correlação negativa entre as funções de avaliação na maioria das funções nos experimentos. Assim, os resultados experimentais mostraram que para o contexto de pesquisa envolvendo as ALPs AGM, BANK, BET e MOM, com as funções de avaliação *DC*, *EC* e *LCC*, são conflitantes. Isso implica que quando uma função é otimizada, a outra conseqüentemente piora. Assim, ao escolher as funções de avaliação, o arquiteto deve priorizar qual objetivo quer otimizar, pois os demais objetivos podem piorar. Se o arquiteto pretende otimizar todas as funções de avaliação, então será preciso selecionar todos os sete objetivos no modelo de avaliação de MOA4PLA.

Outro teste foi aplicado com a intenção de mostrar o efeito das novas funções de avaliação em relação às funções originais. O teste mostrou que em alguns experimentos as

novas funções de avaliação exercem um maior efeito de otimização em relação às originais. A análise da porcentagem de melhoria das novas funções objetivo em relação às funções FM(pla) e CM(pla) mostrou que as funções originais possuem maiores porcentagens de melhoria. Os piores percentuais obtidos pelas novas funções de avaliação reforçam a ideia de que funções mais específicas são mais sensíveis às alterações sofridas pelo projeto de PLA ao longo do processo de otimização.

A fim de aumentar as evidências de correlação entre as funções de avaliação, é indispensável a realização de novos experimentos com outros projetos de ALP para confirmar a correlação das funções de avaliação analisadas. Uma análise qualitativa das soluções encontradas pode permitir identificar por que essas funções estão negativamente correlacionadas. Como trabalho futuro, estudos sobre a depuração das funções serão realizados com o intuito de entender a razão de haver discrepância nos resultados.

Conclusão

A proposta deste trabalho foi refinar o modelo de avaliação da MOA4PLA, propondo novas funções de avaliação, derivadas das funções CM(pla) e FM(pla). As sete novas funções utilizaram as mesmas métricas, separadas por propriedades arquiteturais, com a intenção de se obter avaliações mais precisas sobre as ALPs. Por exemplo, se a intenção é medir acoplamento entre classes, então essa nova função deve trazer informações somente sobre essa propriedade, utilizando métricas que avaliem acoplamento de classes do projeto.

Essas novas funções tratam das seguintes propriedades arquiteturais:

1. **Acoplamento de Componentes;**
2. **Acoplamento de Classes;**
3. **Tamanho de Interfaces;**
4. **Coesão entre Classes;**
5. **Difusão de Características;**
6. **Entrelaçamento de Características;**
7. **Coesão Baseada em Características.**

Depois de definir a proposta, foram implementadas as novas funções na OPLA-Tool, dessa maneira, a ferramenta conta agora com as quatro funções previamente existentes mais as sete novas funções de avaliação.

Depois de implementadas, foi realizado um estudo exploratório com as novas funções de avaliação, no qual o objetivo foi verificar o comportamento das novas funções diante de dois algoritmos multiobjetivos. Os resultados apontaram que não é interessante utilizar muitos objetivos simultaneamente. Assim como mostra a literatura, não é interessante utilizar mais que 4 objetivos na otimização, pois, mais que isso, os resultados não são precisos.

Depois do estudo exploratório, foi realizado um estudo empírico para analisar a correlação entre as novas funções de avaliação. Este estudo realizou-se através de sete combinações com as novas funções de avaliação. Estas combinações foram definidas por funções que tratam do mesmo assunto. Tais estudos mostraram que todas as funções de avaliação são correlacionadas, algumas negativamente e outras positivamente, conforme explicitado no Capítulo 5.

Outro teste foi aplicado com a intenção de mostrar o efeito das novas funções de avaliação em relação às funções originais. O teste evidenciou que as novas funções de avaliação referentes às métricas convencionais exercem um efeito maior do que a função de avaliação CM que agrega todas as métricas convencionais.

O teste de porcentagem de melhoria foi aplicado com a intenção de verificar a melhoria das novas funções de avaliação em relação às funções FM(pla) e CM(pla). Este teste mostrou que as funções originais possuem porcentagens de melhoria maiores que as novas funções de avaliação, reforçando a necessidade de se dispor de funções de avaliação mais específicas e mais sensíveis às alterações realizadas no projeto de PLA.

Uma das contribuições do presente trabalho foi refinar o modelo de avaliação da abordagem MOA4PLA, com as novas funções de avaliação separadas por propriedades arquiteturais. Essas novas funções aumentaram as opções de otimização da abordagem MOA4PLA, e permitem avaliar isoladamente certas propriedades arquiteturais. Além disso, permitem, mostrar o comportamento das funções de avaliação separadas por propriedades arquiteturais.

A outra contribuição foi mostrar a correlação entre as novas funções de avaliação através de sete estudos experimentais. Estes estudos revelam que todas as novas funções são correlacionadas, porém, a grande maioria é correlacionada negativamente.

Pretende-se estender o modelo de avaliação da MOA4PLA utilizando outras métricas diferentes das já implementadas para desenvolver novas funções de avaliação. Dessa maneira, assim como as que foram implementadas neste trabalho, essas novas funções de avaliação mediriam outras propriedades arquiteturais não incluídas no modelo atual. Também como trabalho futuro pretende-se realizar uma análise qualitativa das soluções dos estudos de correlação (vide final Capítulo 5). Além disso, a realização de experimentos

com diferentes combinações de funções de avaliação, agrupando propriedades diferentes com o intuito de descobrir quais conjuntos de funções podem trazer as melhores soluções.

REFERÊNCIAS

ALETI, A.; BUHNOVA, B.; GRUNSKÉ, L.; KOZIOLEK, A.; MEEDEENIYA, I. Software architecture optimization methods: A systematic literature review. *IEEE Transactions on Software Engineering*, v. 39, n. 5, p. 658–683, 2013.

Disponível em <http://dx.doi.org/10.1109/TSE.2012.64>

ARCURI, A.; BRIAND, L. A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. *Software Testing, Verification and Reliability*, v. 24, n. 3, p. 219–250, 2014.

BURKE, E. K.; KENDALL, G. *Search methodologies: Introductory tutorials in optimization and decision support techniques*. Springer, 97–125 p., 2005.

COCHRANE, J. L.; ZELENY, M. *Multiple criteria decision making*. University of South Carolina Press, Columbia, 1973.

COELLO, C. A. C.; LAMONT, G.; VAN VELDHIJZEN, D. *Evolutionary algorithms for solving multi-objective problems*. Genetic and Evolutionary Computation, 2nd ed. Berlin, Heidelberg: Springer, 2007.

COLANZI, T.; VERGILIO, S. A comparative analysis of two multi-objective evolutionary algorithms in product line architecture design optimization. In: *2014 IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2014, p. 681–688.

COLANZI, T. E. *Uma abordagem de otimização multiobjetivo para projeto arquitetural de linha de produto de software*. Tese de Doutorado, Universidade Federal do Paraná, 2014.

COLANZI, T. E.; VERGILIO, S. R.; GIMENES, I. M. S.; OIZUMI, W. N. A search-based approach for software product line design. In: *Proceedings of the 18th International Software Product Line Conference (SPLC 2014)*, 2014.

CONTIERI JUNIOR, A. C.; CORREIA, G. G.; COLANZI, T. E.; DE S. GIMENES, I. M.; JUNIOR., E. A. O.; FERRARI, S.; MASIERO, P. C.; GARCIA, A. F. Extending UML components to develop software product-line architectures: Lessons learned. In: *5th European Conference on Software Architecture (ECSA 2011)*, 2011, p. 130–138.

DARWIN, C. *Origem das espécies, a*. Clássicos de ouro ilustrados. Ediouro Publicações, 2004.

Disponível em <https://books.google.com.br/books?id=WY0wnMq3x9MC>

DEB, K.; JAIN, H. Handling many-objective problems using an improved nsga-ii procedure. In: *IEEE Congress on Evolutionary Computation (CEC 2012)*, pp.1,8, 2012.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182 –197, 2002.

DONEGAN, P. M.; MASIERO, P. C. Design issues in a component-based software product line. In: *Proceedings of Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS)*, 2007, p. 3–16.

FÉDERLE, E. *Uma ferramenta de apoio ao projeto arquitetural de linha de produto de software baseado em busca*. Dissertação de Mestrado, Universidade Federal do Paraná, 2014.

FÉDERLE, E. L.; FERREIRA, T. N.; COLANZI, T. E.; VERGILIO, S. R. OPLA-Tool: A support tool for search-based product line architecture design. In: *Proceedings of the 19th International Conference on Software Product Line, SPLC '15*, New York, NY, USA: ACM, 2015, p. 370–373 (*SPLC '15*,).

GOMAA, H. *Software modeling and design: Uml, use cases, patterns, and software architectures*. Cambridge University Press, 2011.

GUIZZO, G. *Uso de Padrões em Projeto Arquitetural Baseado em Busca de Linha de Produto de Software*. Dissertação de mestrado, Universidade Federal do Paraná, Curitiba, PR, 2014.

GUIZZO, G.; COLANZI, T. E.; VERGILIO, S. R. A Pattern-Driven Mutation Operator for Search-Based Product Line Architecture Design. In: *Proceedings of the 6th Symposium of Search Based Software Engineering (SSBSE)*, Fortaleza: Springer, 2014.

HARMAN, M.; MANSOURI, S. A.; ZHANG, Y. *Search based software engineering: A comprehensive analysis and review of trends techniques and applications*. Relatório Técnico TR-09-03, King's College London, <http://www.dcs.kcl.ac.uk/technical-reports/papers/TR-09-03.pdf>, 2009.

HAUKE, JAN; KOSSOWSKI, T. Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. *Quaestiones Geographicae*, v. 30, n. 2, 2011.

Disponível em <http://www.degruyter.com/view/j/quageo.2011.30.issue-2/v10117-011-0021-1/v10117-011-0021-1.xml>http://geoinfo.amu.edu.pl/qg/archives/2011/QG302{_}087-093.pdf

KANG, K. C.; COHEN, S. G.; HESS, J. A.; NOVAK, W. E.; PETERSON, A. S. *Feature-oriented domain analysis FODA feasibility study*. Relatório Técnico, CMU/SEI-90-TR-21, Carnegie-Mellon University - Software Engineering Institute (SEI), 1990.

KNOWLES, J. D.; CORNE, D. W. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, v. 8, p. 149–172, 2000.

LINDEN, F. J. V. D.; SCHMID, K.; ROMMES, E. *Software product lines in action: The best industrial practice in product line engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

MARTENS, A.; KOZIOLEK, H.; BECKER, S.; REUSSNER, R. Automatically improve software architecture models for performance, reliability, and cost using evolutionary algorithms. In: *Proceedings of the first joint WOSP/SIPEW International Conference on Performance Engineering (WOSP/SIPEW '10)*, New York, NY, USA: ACM, 2010, p. 105–116.

Disponível em <http://doi.acm.org/10.1145/1712605.1712624>

OLIVEIRA JUNIOR, E. A. *System-pla: um método sistemático para avaliação de arquitetura de linha de produto de software baseada em uml*. Tese de Doutorado, Universidade de São Paulo, 2010.

OLIVEIRA JUNIOR, E. A.; GIMENES, I. M. S.; MALDONADO, J. C.; MASIERO, P. C.; BARROCA, L. Systematic evaluation of software product line architectures. *Journal of Universal Computer Science (J.UCS)*, v. 19, p. 25 – 52, 2013.

PARETO, V. *Manuel d'economie politique*. Paris: Ams Press, 1927.

- PRESSMAN, R. S. *Engenharia de software*. 6a. ed. McGraw-Hill, 2006.
- RÄIHÄ, O. A survey on search-based software design. *Computer Science Review*, v. 4, n. 4, p. 203 – 249, 2010.
Disponível em <http://www.sciencedirect.com/science/article/B8JDG-50R0F3S-1/2/9b2de8da263fbf5aebdc1e70f942a1f9>
- SANT'ANNA, C. N. *On the modularity of aspect-oriented design: A concern-driven measurement approach*. Tese de Doutorado, PUC-Rio, Rio de Janeiro/RJ, Brasil, 2008.
- SASTRY, K.; GOLDBERG, D.; KENDALL, G. *Genetic algorithms* Springer, p. 97–125, 2005.
- SCHMID, K.; VERLAGE, M. The economic impact of product line adoption and evolution. *IEEE Software*, v. 19, p. 50–57, 2002.
Disponível em <http://dx.doi.org/10.1109/MS.2002.1020287>
- SEI Arcade Game Maker pedagogical product line. [On-line], <http://www.sei.cmu.edu/productlines/ppl/>.
- SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, p. 591–611, 1965.
- SIMONS, C.; PARMEE, I. Elegant object-oriented software design via interactive, evolutionary computation. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 42, n. 6, p. 1797 –1805, 2012.
- SPLC, S. P. L. C. Product line hall of fame. <Http://splc.net/fame.html>, 2014.
Disponível em <http://splc.net/fame.html>
- VAIDYANATHAN, P. P. *Multirate systems and filter banks*. Prentice-Hall signal processing series. Englewood Cliffs, N.J. Prentice Hall, 1993.
Disponível em <http://opac.inria.fr/record=b1089727>
- VENABLES, W. N.; SMITH, D. M. An introduction to r, ”r manuals, vol, 2. 2012.
- VERGILIO, S.; COLANZI, T.; POZO, A.; ASSUNCAO, W. Search based software engineering: A review from the brazilian symposium on software engineering. In: *Software Engineering (SBES), 2011 25th Brazilian Symposium on*, 2011, p. 50–55.
- WÜST, J. SDMetrics. Disponível em <http://www.sdmetrics.com/>, 2014.

ZITZLER, E.; THIELE, L.; LAUMANN, M.; FONSECA, C. M.; FONSECA, V. G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, p. 117–132, 2003.

A

Artigo publicado no WESB 2015

Otimizando o Projeto de Arquitetura de Linha de Produto de Software com Muitos Objetivos: um Estudo Exploratório

Marcelo C. B. Santos, Wainer Moschetta, Thelma E. Colanzi, Edson Oliveira Jr

Departamento de Informática – Universidade Estadual de Maringá (UEM)
CEP 87.020–900 – Maringá – PR – Brasil

{uemgmarcelo,wainersm}@gmail.com, {thelma,edson}@din.uem.br

Abstract. *The MOA4PLA approach provides search operators and objective functions to optimize and evaluate Software Product Line Architecture (PLA) design by using multi-objective search algorithms (MOAs). However, some of these functions involve metrics with different ranges related to different architectural properties, which affect the function sensibility. This fact motivated an evaluation model refinement, which resulted in six new objective functions. Given the lack of studies on application of MOAs involving many objectives to optimize PLA design, the main goal of this work is presenting an exploratory study about the application of the MOAs NSGA-II and PAES using six objectives to optimize three PLA designs. A behavior analysis of such MOAs in the referred context provided initial evidence that NSGA-II achieved better results than PAES. This work also presents lessons learned from the study carried out.*

Resumo. *A abordagem MOA4PLA propõe operadores de busca e funções objetivo para otimizar e avaliar o projeto de Arquitetura de Linha de Produto de Software (ALP) utilizando algoritmos de busca multiobjetivos (MOAs). No entanto, algumas dessas funções envolvem métricas de diferentes grandezas e referentes a diferentes propriedades arquiteturais, o que influencia na sensibilidade da função. Isso motivou um refinamento do modelo de avaliação originando seis novas funções objetivo. Dada a inexistência de trabalhos nos quais MOAs envolvendo muitos objetivos tenham sido aplicados para otimizar o projeto de ALP, este trabalho tem como principal objetivo apresentar um estudo exploratório no qual os MOAs NSGA-II e PAES foram aplicados usando seis objetivos para otimizar três projetos de ALP. Uma análise do comportamento desses dois MOAs no referido contexto forneceu evidências iniciais de que o NSGA-II alcançou melhores resultados do que o PAES. Este trabalho apresenta também lições aprendidas com a realização do estudo.*

1. Introdução

O objetivo da abordagem de Linha de Produto de Software (LPS) é desenvolver uma família de produtos de software para um determinado domínio com base na reutilização de artefatos de software [Linden et al. 2007]. A Arquitetura de Linha de Produto de Software (ALP) define um projeto comum para os produtos a serem derivados da LPS, por isso é de extrema importância ter uma ALP modular e bem projetada, visando facilitar a manutenção e a evolução da LPS. Uma ALP inclui as características obrigatórias aos produtos da LPS e as características opcionais ou alternativas. Uma característica (*feature*) é uma capacidade do sistema que é relevante e visível para o usuário final.

Modularizar uma ALP não é uma tarefa fácil, pois existem várias alternativas para definir a estrutura dos módulos, as características do produto podem estar distribuídas em

mais de um módulo, há diversos pontos de variação e variantes [Lopez-Herrejon et al. 2005]. Técnicas da Engenharia de Software Baseada em Busca (*Search Based Software Engineering - SBSE*) têm mostrado bons resultados em descobrir soluções próximas das ideais em problemas difíceis como o projeto de ALP [Aleti et al. 2013]. Nesse contexto, foi proposta a abordagem de otimização multiobjetivo denominada *Multi-Objective Approach for Product-Line Architecture Design (MOA4PLA)* [Colanzi et al. 2014]. A MOA4PLA é uma abordagem sistemática e automatizada que usa algoritmos de busca multiobjetivos (MOEAs) para avaliar e melhorar o projeto de ALP.

A MOA4PLA produz um conjunto com as soluções de melhor *trade-off* entre os vários objetivos otimizados. A abordagem inclui um modelo de avaliação para projeto de ALP composto de funções objetivo, as quais são utilizadas para avaliar cada solução gerada no processo de otimização. O modelo de avaliação da MOA4PLA inclui quatro funções objetivo: CM, FM, Ext e Eleg, as quais utilizam métricas de software para medir: princípios básicos de projeto, modularização de características, extensibilidade e elegância do projeto de ALP, respectivamente.

Um estudo empírico realizado por Colanzi et al. (2014) utilizando as funções CM e FM forneceu indícios de que o MOEA NSGA-II (*Non-dominated Sorting Genetic Algorithm*) [Deb et al. 2002] é capaz de obter melhores resultados que o PAES (*Pareto Archive Evolution Strategy*) [Knowles e Corne 2000] em termos de Hipervolume e Distância Euclidiana à solução ideal, quando da aplicação da MOA4PLA para otimizar projetos de ALPs. No entanto, CM e FM envolvem métricas de diferentes grandezas referentes a diferentes propriedades arquiteturais, o que prejudica a sensibilidade da função objetivo em avaliar o projeto de ALP alcançado no processo de busca já que uma métrica com maior grandeza exerce maior influência no valor final da função [Guizzo et al. 2014]. Isso motivou um estudo cujo principal objetivo foi refinar o modelo originalmente proposto em [Colanzi et al. 2014], no qual seis novas funções, separadas por propriedades arquiteturais, foram propostas a partir das funções CM e FM. As novas funções propostas são acrescentadas ao modelo de avaliação original da MOA4PLA, o qual passa a conter as seis novas funções além de Eleg e Ext.

Considerando as funções objetivo derivadas de CM e FM, um novo estudo se faz necessário para avaliar o comportamento de MOEAs, tais como NSGA-II e PAES, durante a otimização de projetos de ALP na presença de muitos objetivos. Os MOEAs possuem algumas limitações quando aplicados a problemas com muitos objetivos, tais como: uma grande parte da população é constituída de soluções não dominadas, a avaliação da diversidade é computacionalmente cara e o cruzamento pode ser ineficiente [Deb e Jain 2012]. Somente estudos envolvendo dois objetivos durante a otimização de projetos de ALP foram encontrados na literatura [Colanzi e Vergilio 2014a, Colanzi et al. 2014, Colanzi e Vergilio 2014b, Guizzo et al. 2014]. Assim, fica evidente a importância de experimentar a aplicação de MOEAs com muitos objetivos para otimizar projetos de ALPs já que esse problema é naturalmente influenciado por vários fatores.

Dessa forma, o objetivo deste artigo é apresentar um estudo exploratório realizado para avaliar o desempenho dos MOEAs NSGA-II e PAES com seis funções objetivo. Para analisar os resultados alcançados, foram utilizados os indicadores de qualidade: Hipervolume e Distância Euclidiana à solução ideal. A partir desse estudo foi possível aprender algumas lições que permitem guiar novos estudos empíricos. Tais lições constituem uma das principais contribuições deste artigo já que até o momento nenhum outro estudo foi publicado envolvendo seis objetivos para a otimização de projetos de ALPs por meio de MOEAs.

Este artigo está organizado em seções. A Seção 2 descreve a abordagem MOA4PLA. A Seção 3 apresenta as funções objetivo propostas e a Seção 4 aborda a descrição do estudo exploratório. As Seções 5 e 6 apresentam os resultados e as lições aprendidas, respectivamente. E, a Seção 7 conclui o artigo e direciona trabalhos futuros.

2. Abordagem MOA4PLA

A MOA4PLA é uma abordagem sistemática e automatizada que usa MOEAs para avaliar e melhorar o projeto de ALP em termos de princípios básicos de projeto, modularização de características e extensibilidade de LPS [Colanzi et al. 2014]. A abordagem utiliza um metamodelo que representa as ALPs e possui operadores de busca específicos para otimizar projetos de ALP, tais como o operador de mutação e o operador de cruzamento dirigidos a características [Colanzi e Vergilio 2014a; Colanzi e Vergilio 2014b]. Após a aplicação dos operadores de busca, cada solução é avaliada de acordo com o modelo de avaliação (vide Seção 2.1) estabelecido pelo arquiteto.

O objetivo da MOA4PLA é a melhoria do projeto de uma ALP, independentemente do MOEA adotado. Por isso, diferentes MOEAs podem ser aplicados no processo de otimização. A Otimização Multiobjetivo consiste em otimizar simultaneamente vários interesses interdependentes e muitas vezes conflitantes e, por isso, não há uma única solução. Assim, o objetivo é encontrar o conjunto de soluções que melhor representa os diversos interesses definidos nos objetivos. Para isso é utilizado o conceito de dominância de Pareto [Pareto 1927] que permite comparar soluções considerando todos os objetivos do problema. As soluções são chamadas de dominadas e não-dominadas. As não-dominadas são as melhores e formam uma aproximação à fronteira de Pareto (PF_{approx}). A união de todos os PF_{approx} , excluindo as soluções dominadas, formam o PF_{known} , que representa as melhores soluções encontradas por um algoritmo para um determinado problema. O PF_{true} representa a fronteira de Pareto ótima e é obtido por meio da união de todos os PF_{known} removendo as soluções dominadas e repetidas, desse modo PF_{true} contém as melhores soluções encontradas para o problema.

Colanzi e Vergilio (2014a) realizaram um estudo envolvendo nove ALPs para comparar o desempenho dos MOEAs NSGA-II e PAES na otimização de projeto de ALP. A principal motivação foi avaliar diferentes estratégias de otimização e, por isso, o NSGA-II foi aplicado em duas versões, sendo uma com cruzamento e outra sem cruzamento, além do PAES que naturalmente não utiliza cruzamento. Nesse estudo, foram utilizados dois objetivos envolvendo a otimização das funções: CM e FM. O NSGA-II obteve melhores resultados em termos de Hipervolume (HV) e de Distância Euclidiana à solução ideal (DE), comparado ao PAES. E a versão do NSGA-II com cruzamento obteve soluções ainda melhores que a versão só com mutação. HV [Zitzler et al. 2003] mede a área de cobertura de uma fronteira de Pareto conhecida (PF_{known}) sobre o espaço de busca. DE mede a distância entre uma solução e a solução ideal em um espaço de objetivos, sendo a solução ideal uma solução com os melhores valores possíveis em todos os objetivos do problema [Cochrane e Zeleny 1973].

2.1 Modelo de avaliação da MOA4PLA

A definição do modelo de avaliação abrange a seleção de métricas e a construção de funções objetivo. No caso da MOA4PLA, o modelo de avaliação inclui métricas convencionais para medir princípios básicos de projeto, tais como acoplamento e coesão, além de métricas específicas para LPS, tais como métricas dirigidas a características e de

extensibilidade de ALP [Colanzi et al. 2014]. A Tabela 1 apresenta uma breve descrição das métricas utilizadas no modelo de avaliação da MOA4PLA.

Tabela 1. Métricas utilizadas no Modelo de Avaliação da MOA4PLA

Atributo	Métrica	Definição
Métricas dirigidas a Características [Sant'Anna 2008] (Função FM)		
Difusão de Características	CDAC	Número de componentes que contribuem para a realização de uma dada característica.
	CDAI	Número de interfaces que contribuem para a realização de uma dada característica.
	CDAO	Número de operações que contribuem para a realização de uma dada característica.
Interação de Características	CIBC	Número de características com os quais uma dada característica compartilha ao menos um componente.
	IIBC	Número de características com os quais uma dada característica compartilha ao menos uma interface.
	OOBC	Número de características com os quais uma dada característica compartilha ao menos uma operação.
Coesão	LCC	Número de características com os quais um dado componente está associado.
Métricas Convencionais [Wüst 2014] (Função CM)		
Coesão Relacional	H	Coesão relacional do projeto por meio do número médio de relacionamentos entre classes dentro de um componente.
Acoplamento	DepPack	Número de pacotes dos quais as classes e interfaces de um dado pacote dependem.
	CDepIn	Número de elementos arquiteturais que dependem dessa classe.
	CDepOut	Número de elementos arquiteturais dos quais essa classe depende.
	DepIn	Número de dependências UML em que o pacote é o fornecedor.
	DepOut	Número de dependências UML em que o pacote é o cliente.
Tamanho	NumOps	Número de operação por interface.
Métrica de Extensibilidade [Oliveira Jr et al. 2013] (Função Ext)		
Extensibilidade	Extens	Soma da capacidade de extensão da LPS em termos de abstração de cada um dos componentes da ALP
Métricas de Elegância [Simons e Parmee 2012] (Função Eleg)		
Elegância	NAC	Elegância de números entre classes, representada pelo desvio padrão dos números de atributos e métodos entre as classes de um projeto.
	EC	Elegância de acoplamentos externos das classes de um projeto
	ATMR	Desvio padrão da razão entre atributos e métodos dentro das classes de um projeto

As métricas para avaliar princípios básicos de projeto são denominadas convencionais. A função objetivo CM é composta pela soma de várias métricas convencionais (apresentadas na Tabela 1). A equação da função é apresentada em Eq. 1, onde c representa o número de componentes, itf representa o número de interfaces e cl representa o número de classes de uma dada ALP. CM utiliza a média de DepPack de todos os componentes da ALP e a média de NumOps de todas as interfaces do projeto.

$$CM(pla) = \sum_{i=1}^c DepIn + \sum_{i=1}^c DepOut + \sum_{i=1}^{cl} CDepIn \sum_{i=1}^{cl} CDepOut + \frac{\sum_{i=1}^c DepPack}{c} + \frac{\sum_{i=1}^{itf} NumOps}{itf} + \frac{1}{\sum_{i=1}^c H} \quad (1)$$

A função FM é usada para avaliar o grau de modularização de uma ALP em termos de características. FM consiste do somatório da soma de cada métrica dirigida a características (Tabela 1). A função FM, proposta por Colanzi et al. (2014), é apresentada a seguir, onde i representa o número de interfaces e f representa o número de características de uma dada ALP.

$$FM(pla) = \sum_{i=1}^c LCC + \sum_{i=1}^f CDAC + \sum_{i=1}^f CDAI + \sum_{i=1}^f CDAO + \sum_{i=1}^f CIBC + \sum_{i=1}^f IIBC + \sum_{i=1}^f OOBC \quad (2)$$

A função objetivo Ext apresenta o valor invertido da métrica Extens. Esse cálculo é feito dessa maneira para que seja possível maximizar a extensibilidade da ALP, uma vez que os objetivos são minimizados na MOA4PLA. A seguir, a função proposta:

$$Ext(pla) = \frac{1}{Extens(pla)} \quad (3)$$

O objetivo da função Eleg é melhorar a elegância do projeto de ALP alcançado em cada solução gerada pela MOA4PLA. Para isso, devem-se minimizar o somatório dos valores das métricas NAC, EC e ATMR conforme apresentado em Eq. 4.

$$Eleg(pla) = NAC(pla) + EC(pla) + ATMR(pla) \quad (4)$$

Como mencionado anteriormente, as funções CM e FM envolvem métricas que tratam de diferentes propriedades arquiteturais e com diferentes grandezas. Resultados experimentais obtidos por Guizzo et al. (2014) mostram que algumas soluções obtidas no processo de otimização tinham melhor acoplamento que outras, porém isso não foi identificado no valor de CM. Logo é preciso dispor de um modelo de avaliação mais preciso, capaz de capturar a qualidade do projeto da ALP. Por isso, foi realizado um refinamento do modelo originalmente proposto por Colanzi et al. (2014). As novas funções objetivo, propostas a partir de CM e FM, são apresentadas na próxima seção.

3. Proposta das Novas Funções Objetivo

As novas funções multiobjetivo foram desenvolvidas utilizando as mesmas métricas utilizadas nas funções CM e FM [Colanzi et al. 2014], porém separadas por propriedades arquiteturais com a intenção de se obter avaliações mais precisas sobre o projeto de ALP. A Tabela 2 apresenta as funções objetivo (seu nome e acrônimo) bem como as métricas utilizadas em cada função.

Tabela 2. Novas Funções Objetivo

Função Objetivo	Métricas
Acoplamento de Componentes (ACOMP)	DepIn, DepOut
Acoplamento de Classes (ACLASS)	CDepIn, CDepOut
Tamanho (TAM)	NumOps
Coesão (COE)	H, LCC
Difusão de Características (DC)	CDAI, CDAO, CDAC
Entrelaçamento de Características (EC)	CIBC, IIBC, OIBC

Três das funções propostas ACOMP, ACLASS e TAM são uma agregação de várias métricas convencionais. DC e EC são uma agregação de várias métricas dirigidas a características. As métricas contam o número de elementos arquiteturais associados a cada característica, avaliam as dependências entre características causadas por características que não estejam bem modularizadas. A função COE avalia a coesão do projeto mesclando a métrica convencional H com a métrica dirigida a características LCC. Abaixo são apresentadas as funções desenvolvidas e a descrição de cada uma:

1. **Acoplamento de Componentes:** composta pelas métricas de dependência de entrada e saída. Esta função mede o acoplamento entre os componentes de uma ALP.

2. **Acoplamento de Classes:** mede o número de elementos arquiteturais que dependem de outras classes do projeto mais o número de elementos dos quais cada classe depende.

$$ACOMP(pla) = \sum_{i=1}^c DepIn + \sum_{i=1}^c DepOut \quad ACLASS(pla) = \sum_{i=1}^{cl} CDepIn + \sum_{i=1}^{cl} CDepOut \quad (5) \quad (6)$$

3. **Tamanho:** composta pela métrica que mede o tamanho de operações de uma classe. Esta métrica mede o número de interfaces de uma classe.

4. **Coesão:** composta por métricas que medem a coesão do projeto de ALP em termos dos relacionamentos internos entre as classes dos componentes e do número de características com os quais cada componente está associado.

$$TAM(pla) = \frac{\sum_{it=1}^{itf} NumOps}{itf} \quad COE(pla) = \sum_{i=2}^c H + \sum_{i=1}^c LCC \quad (7) \quad (8)$$

5. **Difusão de Características:** composta por métricas de difusão de características. Esta função é um somatório de métricas que contam o número de elementos arquiteturais associados a cada característica.

6. **Entrelaçamento de Características:** composta por métricas que analisam a interação entre características. Esta função é a soma dos somatórios de métricas que objetiva avaliar as dependências entre características causadas por alguma característica que não esteja bem modularizada.

$$DC(pla) = \sum_{i=1}^f CDAI + \sum_{i=1}^f CDAO + \sum_{i=1}^f CDAC \quad EC(pla) = \sum_{i=1}^f CIBC + \sum_{i=1}^f IIBC + \sum_{i=1}^f OIBC \quad (9) \quad (10)$$

4. Descrição do Estudo Exploratório

Esta seção descreve o estudo experimental, concentrando-se na definição, planejamento e execução do experimento. Considerando a motivação do estudo apresentada na introdução, o objetivo do experimento é: Analisar os conjuntos de soluções obtidos pelos MOEAs NSGA-II e PAES usando seis objetivos; com o propósito de avaliar a diversidade e o melhor custo benefício (*trade-off*) entre os objetivos do ponto de vista do pesquisador no contexto do projeto de ALP baseado em busca. Somente as novas funções objetivo foram utilizadas no estudo. As variáveis independentes são NSGA-II e PAES, e as dependentes são o *fitness* das soluções, o melhor *trade-off* entre os objetivos e a diversidade das soluções. Os indicadores de qualidade Hipervolume (HV) e Distância Euclidiana à solução ideal (DE) foram usados para medir, respectivamente, a diversidade das soluções e o *trade-off* entre os objetivos. . O teste de hipótese de Wilcoxon foi utilizado com significância de 95% (p-value $\leq 0,05$), a fim de verificar a diferença estatística entre os MOEAs com relação à diversidade de soluções.

A hipótese nula do estudo é a hipótese H0, em que não existe diferença significativa entre o PAES e o NSGA-II considerando Hipervolume (HV) e Distância Euclidiana (DE). As duas hipóteses alternativas do estudo são H1 e H2. H1 assume que o PAES é melhor em termos de Hipervolume e Distância Euclidiana que o NSGA-II. Na hipótese H2, o NSGA-II é melhor que o PAES. Essas hipóteses são definidas abaixo em termos dos indicadores utilizados na comparação: HV e DE.

$$\begin{aligned} H_0: & HV_{NSGA-II} = HV_{PAES} \text{ and } DE_{NSGA-II} = DE_{PAES} \\ H_1: & HV_{NSGA-II} < HV_{PAES} \text{ and } DE_{NSGA-II} > DE_{PAES} \\ H_2: & HV_{NSGA-II} > HV_{PAES} \text{ and } DE_{NSGA-II} < DE_{PAES} \end{aligned}$$

4.2 ALPs Utilizadas no Estudo

O experimento envolveu três ALPs, cujas principais informações são apresentadas na Tabela 3. Todas as ALPs são baseadas em componentes e seguem o estilo em camadas. Arcade Game Maker (AGM) é uma LPS criada pelo SEI que inclui três jogos de arcade. Duas versões dessa ALP foram utilizadas: AGM-1 [SEI 2015] e AGM-2 [Contieri Jr et al. 2011]. A Mobile Media (MOM), versão MOM-2 [Contieri Jr et al. 2011], é uma LPS que apoia o gerenciamento de diferentes tipos de mídias.

Observa-se que o valor da função objetivo ACLASS é zero. Isto se justifica pela forma de medição das métricas CDepIn e CDepOut que contam o número de elementos que dependem de uma determinada classe e dos quais essa classe depende, respectivamente, por meio de dependências UML ou dependências de uso. As ALPs utilizadas não possuem esses tipos de dependências, o que justifica os valores zerados.

Tabela 3. Informações sobre os projetos de ALP utilizados no estudo

ALP	# Componentes	# Interfaces	# Classes	# Variabilidades	# Características	<i>Fitness Original (ACOMP, ACLASS, TAM, COE, DC, EC)</i>
AGM-1	9	14	20	4	9	(28; 0; 35,71; 25,25; 300; 66)
AGM-2	9	14	21	5	11	(31; 0; 3,93; 29,25; 359; 112)
MOM-2	8	13	10	7	13	(27; 0; 6,153; 22,33; 405; 107)

4.3. Operação e Execução do estudo

A OPLA-Tool [Colanzi et al. 2014], ferramenta que apoia a aplicação da MOA4PLA, foi utilizada na execução do experimento. Os MOEAs NSGA-II e PAES foram executados 25 vezes para cada ALP. Os algoritmos foram configurados usando os mesmos parâmetros adotados por Colanzi e Vergilio (2014a). Esses parâmetros são: tamanho da população= 100, taxa de mutação= 0,9 e número de gerações= 300. Nenhum operador de cruzamento foi aplicado no NSGA-II por restrições da ferramenta.

4.4. Ameaças à validade

A principal questão que se deve levar em conta como um risco que pode afetar a validade do experimento é o tamanho da amostra e o tamanho das ALPs utilizadas já que foram utilizados somente três projetos de ALPs acadêmicas e, com ALPs maiores, a solução do problema tende a ficar mais complexa. Todas as métricas utilizadas nas funções objetivo e, por consequência, no experimento, já foram utilizadas em estudos anteriores [Colanzi e Vergilio 2014a, Colanzi et al. 2014, Guizzo et al. 2014]. Outra possível ameaça está relacionada com a aleatoriedade dos algoritmos de busca. Para mitigar essa ameaça, cada algoritmo foi executado 25 vezes, pois isso garante resultados consistentes mesmo com resultados diferentes a cada execução. Além disso, foram adotados os mesmos parâmetros de configuração adotados por Colanzi e Vergilio (2014a). Todas essas ameaças podem ser tratadas em estudos futuros.

5. Análise e Interpretação dos Resultados

Esta seção apresenta os resultados obtidos com a realização do experimento. A Tabela 4 apresenta o número de soluções que formam os conjuntos PF_{true} e PF_{known} para cada ALP. Observa-se que o NSGA-II encontrou um número menor de soluções que o PAES, porém, escolher manualmente um projeto de ALP dentre um conjunto numeroso de soluções encontradas é uma tarefa árdua para o arquiteto. Nota-se que o PF_{true} é formado por soluções encontradas pelos dois algoritmos. O NSGA-II demorou em média 10 minutos em cada execução, enquanto o PAES demorou cerca de 120 minutos.

Na Tabela 5 pode-se observar os valores de *fitness* e de DE das soluções com menor DE encontradas pelo NSGA-II e pelo PAES, além do *fitness* da solução ideal. Os valores de *fitness* são apresentados na seguinte ordem: ACOMP, ACLASS, TAM, COE, DC e EC. Pode-se notar que as soluções encontradas pelo NSGA-II (valores destacados em negrito na tabela) são melhores que as encontradas com o PAES já que têm menor DE em relação à solução ideal. A solução ideal é composta pelos valores mínimos de cada função objetivo alcançados nas soluções que compõem o PF_{true} .

Os valores médios de hipervolume, calculados com base nos resultados de cada uma das 25 execuções de cada um dos MOEAs, podem ser vistos na Tabela 6. As funções objetivo foram normalizadas para o cálculo do hipervolume. Pode-se constatar que os valores de hipervolume do NSGA-II também foram melhores do que os valores do PAES. Os maiores valores de hipervolume estão destacados em negrito.

Tabela 4: Número de Soluções Encontradas

ALP	PF _{true}	PF _{known} do NSGA-II	PF _{known} do PAES
AGM-1	471	52	423
AGM-2	811	161	657
MOM-2	696	119	577

Tabela 5: Soluções de Menor Distância Euclidiana à Solução Ideal

ALP	Solução Ideal (ACOMP, ACLASS, TAM, COE, DC, EC)	NSGA-II		PAES	
		DE	Fitness da solução	DE	Fitness da solução
AGM-1	(15; 0; 1,09; 15,1; 220; 46)	8,8	(17; 0; 3,8; 22,2; 223; 49)	226,0	(27; 0; 2,1; 43,0; 430; 124)
AGM-2	(18; 0; 1,08; 23,1; 262; 74)	8,5	(22; 0; 3,6; 27,1; 265; 79)	199,0	(24; 0; 3,2; 38,1; 455; 120)
MOM-2	(15; 0; 1,16; 17,5; 340; 70)	13,7	(19; 0; 5,7; 20,2; 345; 81)	419,6	(45; 0; 1,9; 56,0; 740; 187)

Tabela 6: Hipervolume

ALP	NSGA-II		PAES		p-value
	Média	Desvio Padrão	Média	Desvio Padrão	
AGM-1	0,00338	0,00734	0,00173	0,00084	8,14507
AGM-2	0,00408	0,00031	0,00139	0,00105	9,31647
MOM-2	0,00369	0,00031	6, 16777	0,00042	1,14507

Após o cálculo dos indicadores de qualidade apresentados anteriormente, o teste de normalidade Shapiro-Wilk foi aplicado sobre os resultados de HV cujo resultado aponta que os dados não seguem uma distribuição normal. Todos os resultados obtiveram relevância estatística (95% de confiança no teste). Então, foi aplicado o teste de Wilcoxon para avaliar a validade da hipótese nula adotada no presente trabalho. Os resultados apontam que existe diferença entre os conjuntos de dados encontrados pelos dois algoritmos com 95% de confiança (vide valores de *p-values* na Tabela 6). A análise dos *boxplots* permite constatar a consistência dos dados obtidos pelos MOEAs bem como qual deles obteve a maior mediana em termos de HV. Como é possível observar na Figura 2, o NSGA-II alcançou a maior mediana para os três conjuntos de dados.

Com base na média de HV e na análise dos *boxplots*, observa-se que o NSGA-II encontra maior diversidade de soluções do que o PAES. Como apresentado anteriormente, NSGA-II também encontra soluções com menores DE do que PAES e, portanto, consegue otimizar mais os objetivos definidos para o processo de otimização. Portanto, com nível de significância de 95% a hipótese nula foi rejeitada. Ainda, o NSGA-II é melhor que PAES e, portanto, fornece indícios de que a hipótese alternativa H2 deve ser aceita, a qual afirma que o NSGA-II é melhor que PAES em termos de HV e DE. Apesar disso, vários pontos foram observados durante este estudo no que diz respeito à aplicação dos MOEAs com muitos objetivos. A partir dessa observação algumas lições foram aprendidas, as quais são abordadas na próxima seção.

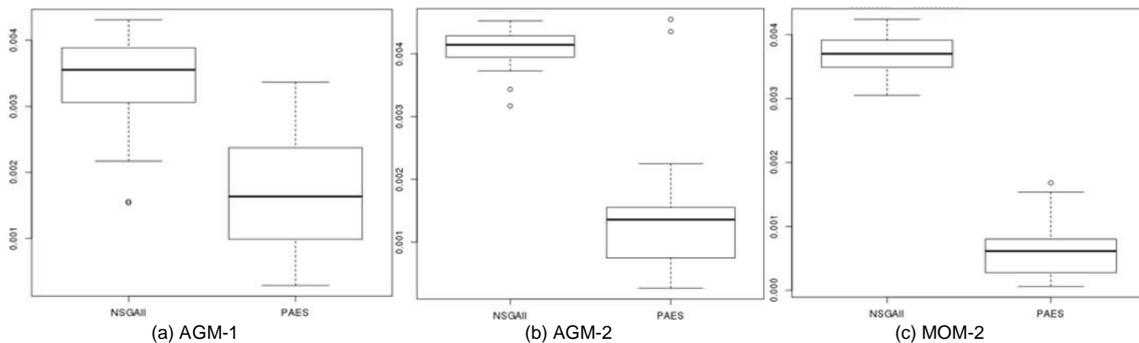


Figura 2: Boxplots dos resultados de hipervolume

6. Lições Aprendidas

A seguir são apresentadas as lições aprendidas a partir da condução do experimento. Essas lições servem como base para a realização de futuros estudos.

Número de Soluções Encontradas: O tamanho dos conjuntos de soluções encontrados pelos dois MOEAs fornece evidências de que ambos não conseguem resolver o problema de projeto de ALP na presença de seis objetivos de forma satisfatória do ponto de vista do arquiteto, já que encontram muitas soluções, confirmando as evidências relatadas por Deb e Jain [2012]. Nesse caso, seria melhor conduzir experimentos com um número menor de objetivos.

Tempo de Execução: Considerando que o NSGA-II teve um tempo de execução médio de 10 minutos e o PAES de 120 minutos, pode-se dizer que o tempo de execução do NSGA-II na presença de muitos objetivos é mais viável.

Otimização das Funções: Uma comparação entre os valores de *fitness* das soluções de menor DE (Tabela 5) com os valores de *fitness* originais (Tabela 3) permitiu observar que o NSGA-II conseguiu otimizar grande parte dos objetivos. Por exemplo, o valor original da função ACOMP para a ALP AGM-1 era 28,0. A solução de menor DE encontrada pelo NSGA-II diminuiu o valor do acoplamento entre componentes para 17,0 enquanto o PAES somente reduziu o valor em 1 unidade (27,0). Em outros casos, como o da MOM-2, o NSGA-II conseguiu otimizar o valor enquanto o PAES obteve uma solução de pior valor que o original para essa função. O mesmo ocorreu com as funções de coesão (COE), difusão e entrelaçamento entre características (funções DC e EC). Por outro lado, o PAES conseguiu obter melhores resultados que o NSGA-II em todas as ALPs ao otimizar o tamanho das interfaces do projeto (função TAM). Considerando o acoplamento de classes (ACCLASS) os algoritmos não conseguiram otimizar o valor da função uma vez que os valores originais de todas as ALPs eram zero. Isso indica que para estas ALPs a função ACCLASS não tinha o que otimizar.

Soluções Próximas do Ideal: O NSGA-II conseguiu encontrar soluções próximas da solução ideal mesmo diante do desafio de otimizar seis objetivos simultaneamente.

Número de Funções: Apesar de usar seis objetivos, o NSGA-II conseguiu valores de *fitness* melhores que os originais. Isso evidencia tanto a viabilidade de aplicação das funções propostas como a sua sensibilidade às mudanças nas propriedades arquiteturais que elas pretendem medir.

Pode-se concluir que o NSGA-II consegue encontrar melhores projetos de ALP do que os projetos originais, porém ele não consegue explorar adequadamente o espaço de busca diante de muitos objetivos. Desse modo, parece ser mais promissor o uso de um número menor de funções objetivo em se tratando da otimização de ALPs.

7. Considerações Finais

Neste estudo foram relatados os resultados de uma avaliação do comportamento dos MOEAs NSGA-II e PAES diante de muitos objetivos. Para isso, utilizou-se a abordagem multiobjetivo MOA4PLA. Os resultados fornecem indícios de que NSGA-II supera PAES para resolver problemas com seis objetivos considerando as ALPs utilizadas. No entanto, o desempenho do NSGA-II ainda assim não é adequado, pois produz uma grande quantidade de soluções, dificultando a seleção de um projeto de ALP por parte do arquiteto. Nesse sentido, pode-se concluir que é viável a utilização das novas funções

propostas neste trabalho, porém, seria melhor utilizá-las em conjuntos variados de funções objetivos utilizando um menor número de objetivos.

Como trabalhos futuros serão realizados estudos utilizando as funções propostas com um número menor de objetivos a serem otimizados simultaneamente, além de uma análise qualitativa das soluções conduzida com arquitetos.

Agradecimentos

Os autores agradecem ao DInf/UFPR e ao Giovani Guizzo pelo apoio na execução dos experimentos. Agradecem também à CAPES pelo apoio financeiro.

Referências

- Aleti, A.; Buhnova, B.; Grunske, L.; Koziolok, A. and Meedeniya I. (2013). "Software architecture optimization methods: A systematic literature review". IEEE Trans. on Software Engineering, 39(5):658–683.
- Cochrane, J. L. and Zeleny, M. (1973). "Multiple Criteria Decision Making". University of South Carolina Press, Columbia.
- Colanzi, T. E. e Vergilio, S. R. (2014a) "A Comparative Analysis of Two Multi-Objective Evolutionary Algorithms in Product Line Architecture Design Optimization" In: ICTAI, pp.681-688.
- Colanzi, T. E. e Vergilio, S. R. (2014b) "A Feature-Driven Crossover Operator for Product Line Architecture Design Optimization". In: Computer, Software and Applications Conference, pp.43-52.
- Colanzi, T. E.; Vergilio, S. R.; Gimenes, I. and Oizumi, W. N. (2014) A search-based approach for software product line design. In: Int. Software Product Line Conference, pp. 237-241, Florence, Italy.
- Contieri Jr, A. C.; Correia G. G.; Colanzi, T. E.; Gimenes, I.; Oliveira Jr, E.; Ferrari, S.; Masiero, P. C.; Garcia, A. F. (2011) "Extending UML Components to Develop Software Product-Line Architectures: Lessons Learned", In: European Con. on Software Architecture, pp. 130-138, Essen, Alemanha.
- Deb, K.; Pratap, A.; Agarwal. and Meyarivan T. (2002) "A fast and elitist multiobjective genetic algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation. 6(2):182-197.
- Deb, K. and Jain, H. (2012) "Handling many-objective problems using an improved NSGA-II procedure", In: IEEE Congress on Evolutionary Computation (CEC 2012), pp.1,8.
- Guizzo, G.; Colanzi, T. E. and Vergilio, S. R. (2014) "Applying Design Patterns in Search-Based Product Line Architecture Design," In: SSBSE, pp 77-91, Fortaleza, Brasil.
- Knowles, J. D. and Corne, D. W. (2000). "Approximating the non-dominated front using the Pareto archived evolution strategy". Evolutionary Computation, 8:149–172.
- Linden, F.; Schmid, K. and Rommes, E. (2007) "Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering". Springer, 2007.
- Lopez-Herrejon, R. E.; Batory D. and Cook, W. (2005). "Evaluating support for features in advanced modularization technologies". In: European Conf. on Object-Oriented Programming, pp. 169–194.
- Oliveira Jr, E.; Gimenes, I. ; Maldonado, J. C.; Masiero, P. C.; Barroca, L. (2013). "Systematic evaluation of software product line architectures". Journal of Universal Computer Science, 19:25 – 52.
- Pareto, V. (1927). "Manuel D'Economie Politique". Ams Press, Paris.
- Sant'Anna, C. N. (2008). "On the Modularity of Aspect-Oriented Design: A Concern-Driven Measurement Approach". PhD thesis, PUC-Rio, Rio de Janeiro/RJ, Brasil.
- SEI (2015). Arcade Game Maker Pedagogical Product Line. in: <http://www.sei.cmu.edu/productlines/ppl/>
- Simons, C. L. and Parmee, I.C. (2012) "Elegant object-oriented software design via interactive, evolutionary computation". IEEE Trans. Systems, Man and Cybernetics, 42(6):1797-1805.
- Zitzler, E. and Thiele, L., Laumanns, M., Fonseca, C. M. e Fonseca, V. G. (2003) "Performance assessment of multiobjective optimizers: An analysis and review". IEEE Transactions on Evolutionary Computation, 7:117-132.
- Wust, J. (2014). SDMetrics. in: <http://www.sdmetrics.com/>.

B

Algumas PLAs obtidas nos estudos de correlação

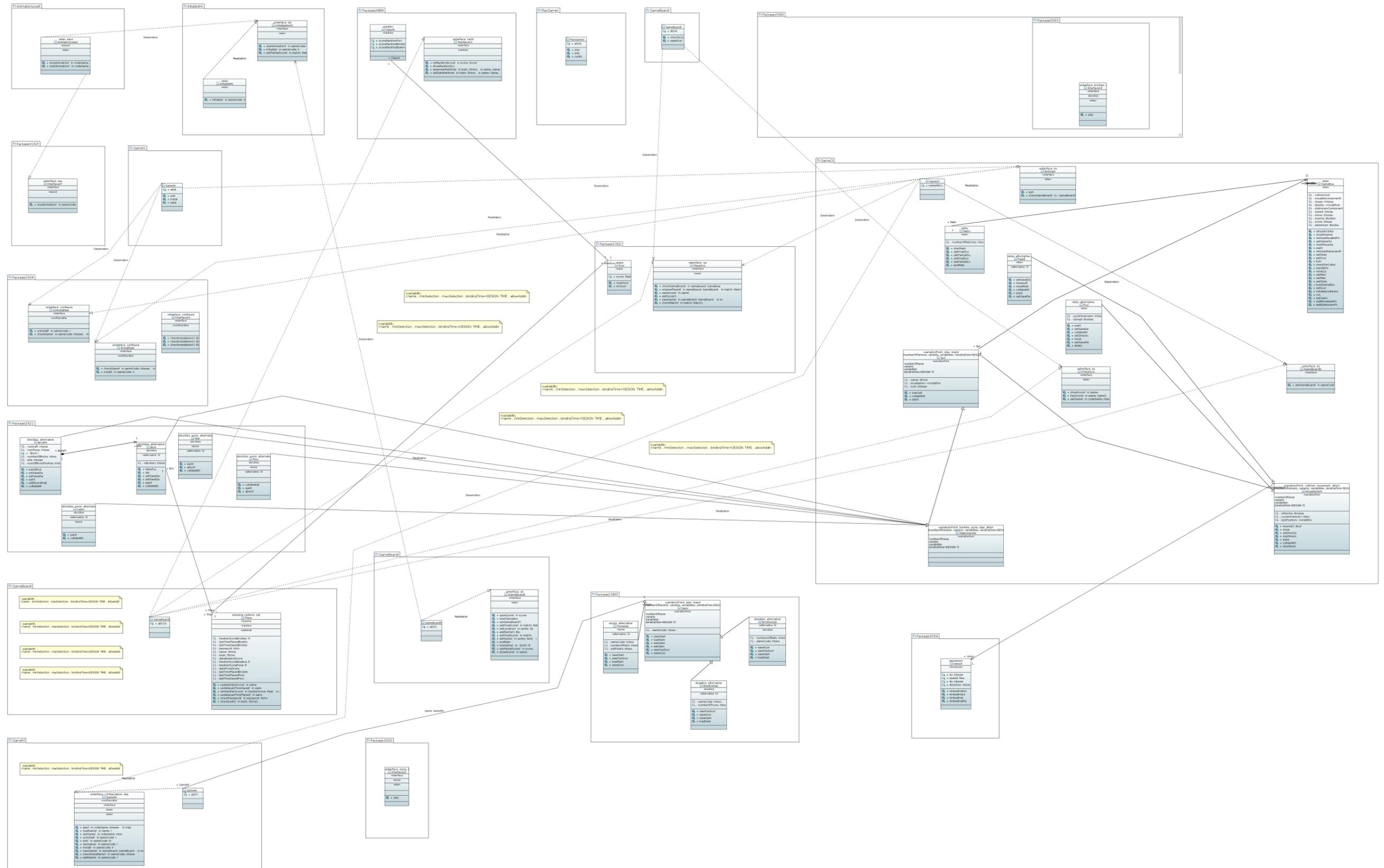


Figura B.1: Projeto de PLA da AGM obtido no Experimento 1

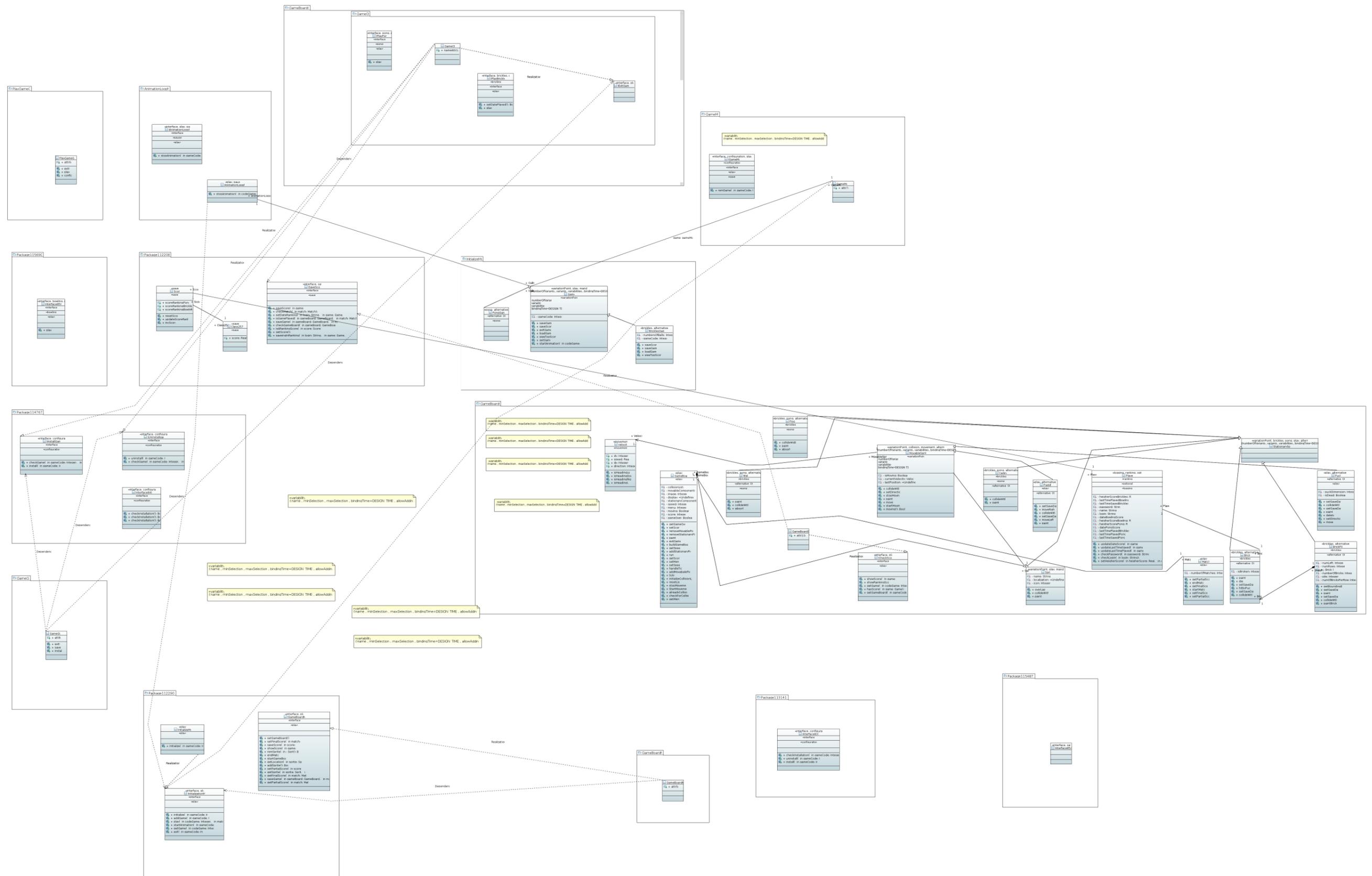


Figura B.2: Projeto de PLA da AGM obtido no Experimento 2

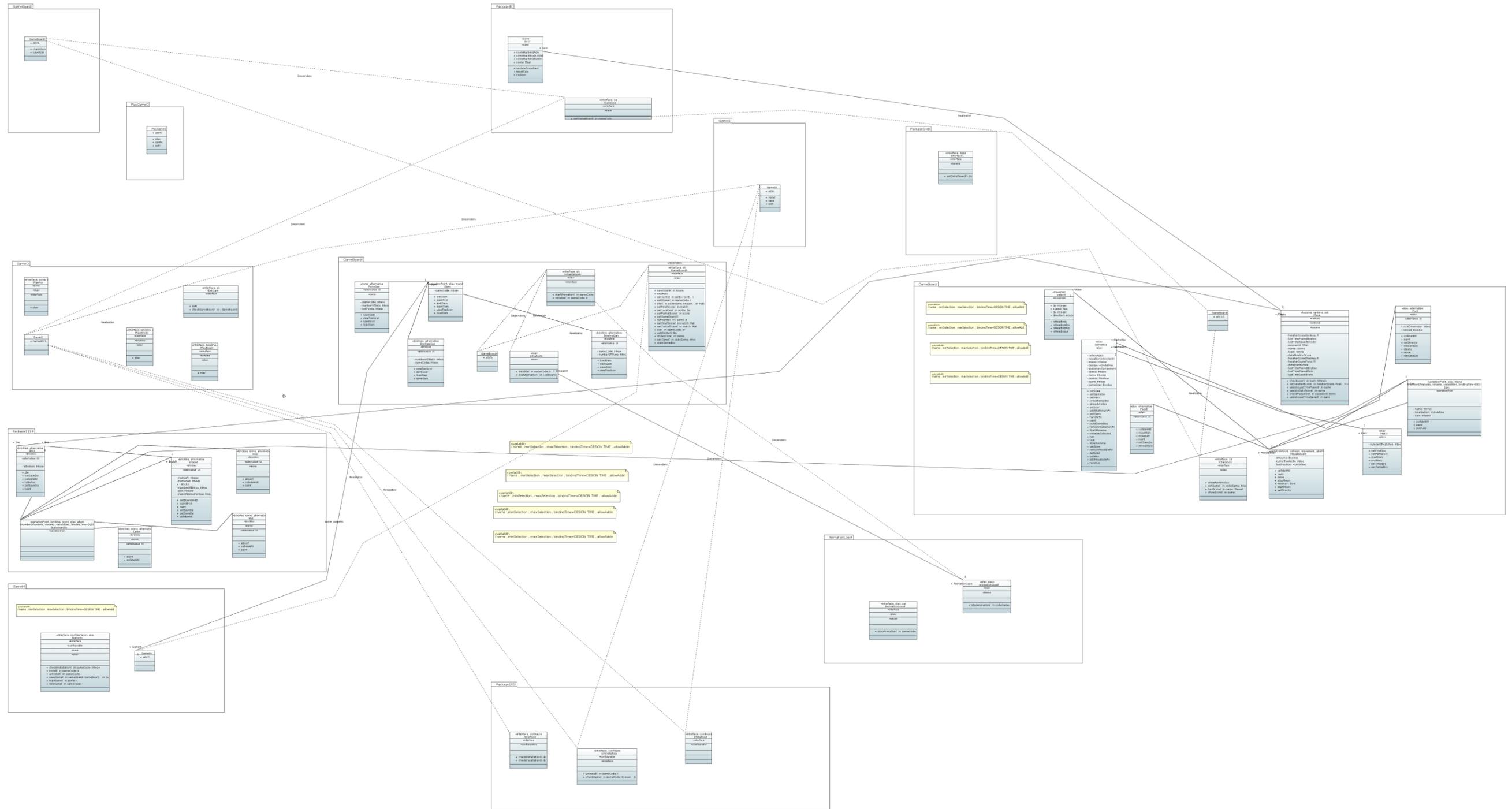


Figura B.3: Projeto de PLA da AGM obtido no Experimento 3

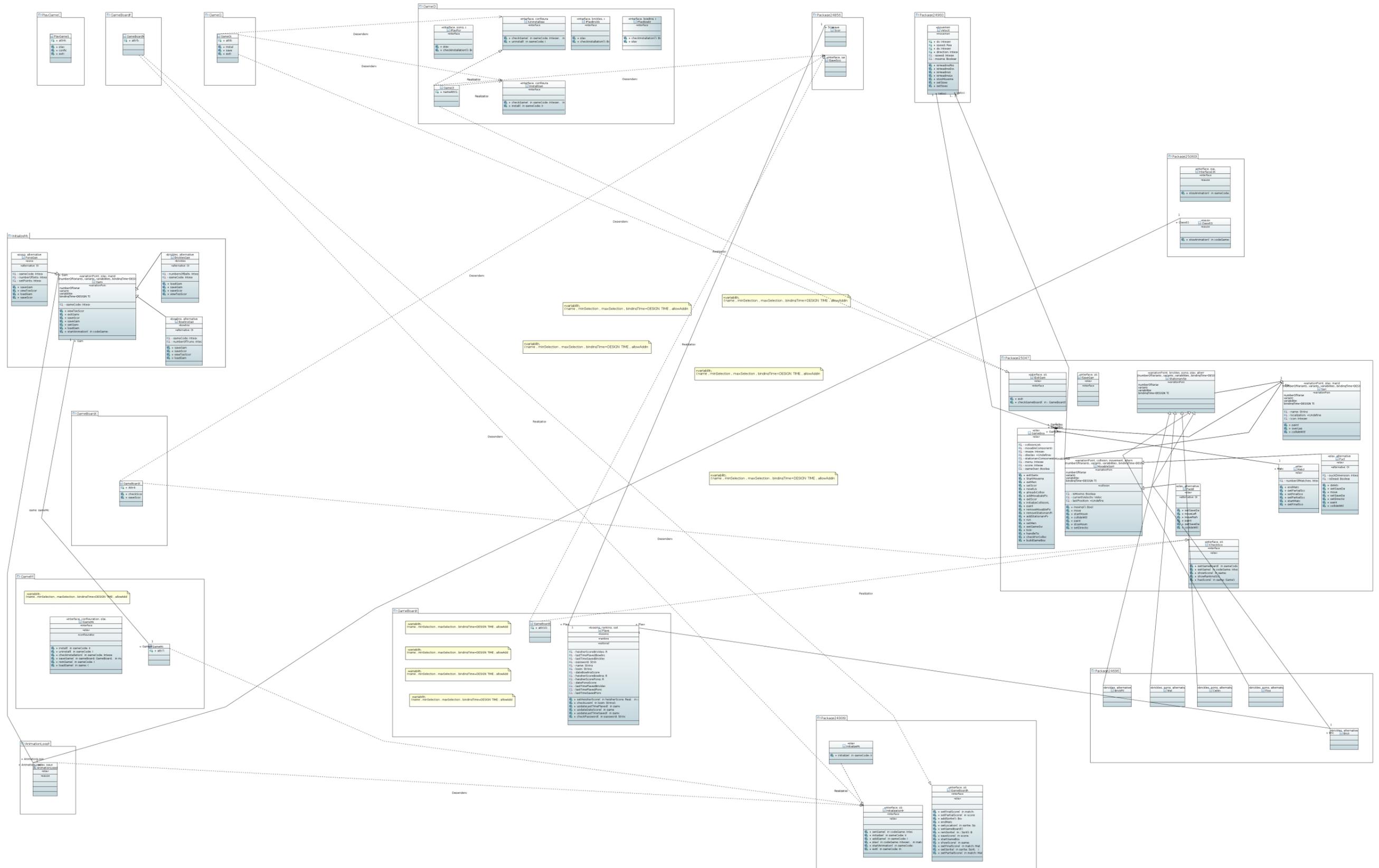


Figura B.6: Projeto de PLA da AGM obtido no Experimento 6

