

## AVALIAÇÃO DA OTIMIZAÇÃO DE ARQUITETURAS DE LINHA DE PRODUTO DE SOFTWARE DE MÉDIO PORTE UTILIZANDO A OPLA-TOOL

Rodrigo Gidioni Wagatsuma da Silva, Thelma Elita Colanzi, e-mail:  
[teclopes@din.uem.br](mailto:teclopes@din.uem.br)

Universidade Estadual de Maringá / Centro de Tecnologia, PR.

**Ciências Exatas e da Terra /Metodologia e Técnicas da Computação**

**Palavras-chave:** OPLA-Tool, Algoritmos genéticos, Linha de produto de Software

### RESUMO

A abordagem *Multi-Objective Approach for Product-Line Architecture Design* (MOA4PLA) é reconhecida por seus resultados promissores na otimização de arquiteturas de linha de produto de software (PLA) por meio de algoritmos de busca multiobjetivos. A ferramenta OPLA-Tool, que automatiza essa abordagem, exige uma cuidadosa configuração de parâmetros pelo usuário para alcançar os melhores resultados. Estudos anteriores exploraram como ajustar a OPLA-Tool para otimização de PLAs de pequeno porte. E, em um projeto anterior foi desenvolvida a PLA de médio porte denominada BET2, mas ela ainda não havia sido otimizada na OPLA-Tool. Neste contexto, o objetivo deste trabalho foi realizar novos estudos para avaliar empiricamente como configurar a OPLA-Tool para PLAs de médio porte a fim de instruir melhor os usuários da ferramenta em direção à sua adoção na indústria. Os resultados dos experimentos foram avaliados com base em indicadores de qualidade de algoritmos multiobjetivos que permitiu identificar a melhor configuração de parâmetros do algoritmo para otimizar PLA de médio porte, corroborando os resultados obtidos em pesquisas anteriores sobre o uso da OPLA-Tool.

### INTRODUÇÃO

A abordagem de Linha de Produto de Software (LPS) visa promover a reutilização de artefatos de software já desenvolvidos em sistemas do mesmo domínio, o que, como resultado, reduz o tempo voltado para o desenvolvimento e minimiza os custos da produção de software. Dessa forma, uma LPS contém um conjunto de funcionalidades comuns e que satisfazem as necessidades de um segmento de mercado particular.

O principal artefato reutilizado é a PLA (*Product Line Architecture*) e, para isso, Colanzi (2014) propôs a MOA4PLA (*Multi-Objective Approach for Product Line Architecture Design*), uma abordagem sistemática e automatizada que utiliza de um

algoritmos genéticos multiobjetivos para efetuar a otimização de uma arquitetura, visto que a otimização de uma PLA exige muito esforço do arquiteto de software. A MOA4PLA avalia e melhora o projeto de PLA em termos de propriedades básicas de projeto, modularização de características e extensibilidade de LPS.

Algoritmos Genéticos Multiobjetivos, como o NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) (DEB et al., 2002), utilizado nesse projeto, otimizam mais de uma função objetivo simultaneamente, o que atende às necessidades do projeto de PLA que sofre influência de vários fatores. Freire et al. (2020) identificaram os melhores parâmetros para configurar a OPLA-Tool PLAs de pequeno porte, a saber: taxa de cruzamento, taxa de mutação, operadores de cruzamento e mutação, número de gerações, tamanho da população e número de avaliações de fitness. Okada et al. (2022) utilizaram as mesmas configurações de Freire et al. (2020) para rodar os estudos experimentais envolvendo a PLA de médio porte BET, alcançando evidências sobre valores de parâmetros a serem configurados para PLAs de médio porte. Nesse contexto, o presente estudo teve como objetivo investigar como configurar a OPLA-Tool para a PLA BET2 e avaliar as soluções obtidas.

## MATERIAIS E MÉTODOS

Nesse sentido, a fim de alcançar o objetivo do projeto, foram executados experimentos com a PLA BET2, desenvolvida no trabalho de Okada et al. (2022), seguindo as mesmas diretrizes dos estudos anteriores utilizando a OPLA-Tool. Foram testadas quatro configurações, as quais estão descritas na Tabela 1, sendo que cada uma delas foi executada 30 vezes, utilizando-se dos servidores da UEM. As funções objetivo utilizadas reference-se a modularização de características (FM), coesão relacional (COE) e acoplamento de classes (ACLASS).

Após finalizada a execução de cada configuração, executou-se a normalização dos dados. Para isso, como a análise dos resultados seria feita de forma quantitativa, foi preciso pegar o valor de *fitness* de cada solução e então colocá-los em uma planilha com um algoritmo desenvolvido para normalizar as saídas das configurações. O valor de *fitness* é um conjunto de três valores que representa cada uma das funções objetivo e, junto com uma representação gráfica, caracteriza cada uma das soluções.

Desse modo, visando avaliar a qualidade de conjuntos de soluções em problemas com múltiplos objetivos, os valores normalizados foram utilizados para calcular o *hypervolume*, que é uma métrica usada para quantificar quão boas são as soluções de um determinado problema. A partir disso, o *boxplot* foi a ferramenta gráfica escolhida para representar visualmente a distribuição do conjunto de dados, que serviu para que uma primeira comparação fosse feita, a qual foi posta à prova pelos testes estatísticos descritos a seguir.

1. **Shapiro-Wilk:** Para verificar a confiabilidade da análise gráfica obtida pelo *boxplot*, a primeira técnica estatística usada foi o teste de Shapiro-Wilk, o qual teve como objetivo avaliar a normalidade dos dados. Como os valores eram não normais e não atendiam às suposições necessárias para Análise de Variância (ANOVA), determinou-se que seria efetuado o teste não paramétrico de Kruskal-Wallis.

2. **Kruskal-Wallis:** Comumente usado para comparação de três ou mais configurações quando os valores não seguem uma distribuição normal, o teste estatístico de Kruskal-Wallis foi o segundo a ser utilizado para determinar se havia diferenças estatisticamente significativas entre as medianas dos grupos. Desse modo, como houveram diferenças estatísticas, Vargha-Delaney foi o teste utilizado para verificar o quão grande eram essas diferenças.

3. **Vargha-Delaney:** Utilizado para quantificar a magnitude da diferença estatística entre os grupos, o teste de Vargha-Delaney mediu o tamanho do efeito (*effect size*) para que se pudesse concluir qual era a melhor configuração.

**Tabela 1:** Configurações dos experimentos realizados com a OPLA-Tool

Config.	Tamanho da População	# de Gerações	# Avaliações de Fitness	Operadores de Cruzamento	Operadores de Mutação	TC	TM
1	100	300	30000	FDC+CC	ALL	0.4	0.8
2	100	300	30000	FDC+MC	ALL	0.4	0.8
7	200	15	3000	MC+CC	ALL	0.4	0.9
8	100	300	30000	MC+CC	ALL	0.4	0.8

Legenda: TC=Taxa de cruzamento TM=Taxa de Mutação

## RESULTADOS E DISCUSSÃO

A análise visual do *boxplot* presente na Figura 1 sumariza o resultado avaliado pelos testes estatísticos. No *boxplot*, a configuração 1 está denominada como exp1, a configuração 2 denominada como exp2, a configuração 7 denominada como exp7 e a configuração 8 denominada de exp8. A configuração 8 aparentava possuir a melhor solução uma vez que para o indicador de qualidade hypervolume, quanto mais próximo de um, melhor o resultado.

O teste de Shapiro-Wilk mostrou que todas as configurações tinham distribuição não-normal. Dessa forma, o teste de Kruskal-Wallis serviu para verificar que haviam diferenças estatisticamente significativas entre os resultados das quatro configurações. Por fim, o teste de Vargha-Delaney, ao medir o tamanho do efeito, atestou que a configuração 8 é a melhor, assim como previsto pelo *boxplot*.

## CONCLUSÕES

Após a análise dos resultados, gráficos de *boxplots* e testes estatísticos realizados foi possível concluir que, sem alteração nos outros indicadores, os

operadores de cruzamento Modular Crossover (MC) e Complementary Crossover (CC) são determinantes para encontrar os melhores resultados para a LPS-BET 2, resultado previsto no trabalho de Da Silva et al. (2022). Sendo que a configuração 8 apresentou a melhor combinação de indicadores e alcançou o melhor resultado.

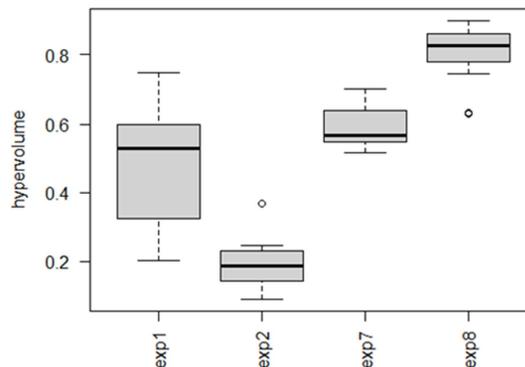


Figura 1: Boxplot dos experimentos executados

## AGRADECIMENTOS

Agradecemos ao CNPq pelo apoio financeiro, ao Willian M. Freire pelo auxílio na realização do projeto.

## REFERÊNCIAS

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

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.

FREIRE, W. M., MASSAGO, M., ZAVADSKI, A. C., AMARAL, A. M. M. M., COLANZI, T. E.. OPLA-Tool v2.0: a Tool for Product Line Architecture Design Optimization. *In*: XXXIV Brazilian Symposium on Software Engineering, **Anais [...]**, New York: ACM, 2020, p. 818–823.

OKADA, L. F. ; COLANZI, THELMA E. . Avaliação Quanti-Qualitativa de Arquiteturas de Linha de Produto de Software Otimizadas utilizando a OPLA-Tool. *In*: 30.o Encontro Anual de Iniciação Científica e 10.o Encontro Anual de Iniciação Científica Júnior, 2021, Maringá. **Anais [...]**, 2021.

DA SILVA, D.F., OKADA, L.F., ASSUNÇÃO, W.K.G., COLANZI, T. E.. Intensifying the search-based optimization of product line architectures with crossover operators. **Empirical**

33° Encontro Anual de Iniciação Científica  
13° Encontro Anual de Iniciação Científica Júnior



10 e 11 de Outubro de 2024

**Software Engineering** 27, 166 (2022).

