

Apoiando a Adoção da OPLA-Tool por usuários novatos

João Pedro Zen Sirino(PIBIC/CNPq/FA/Uem), Luiz Fernando Okada(PIBIC/CNPq/FA/Uem), Thelma Elita Colanzi (Orientadora), Aline M. M. M. Amaral (Coorientadora), e-mail: ra124360@uem.br.

Universidade Estadual de Maringá / Centro de Tecnologia, PR.
Ciência da Computação / Engenharia de Software

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

Resumo:

A *Multi-Objective Approach for Product-Line Architecture Design* (MOA4PLA) é uma abordagem seminal que tem alcançado resultados promissores na otimização de projeto de arquitetura de linha de produto de software (PLA) utilizando algoritmos de busca. A ferramenta OPLA-Tool foi desenvolvida com o objetivo de automatizar o uso da MOA4PLA. Porém, como a OPLA-Tool utiliza algoritmos de busca, há uma série de parâmetros que devem ser configurados pelo usuário a fim de obter os melhores resultados possíveis. Isso dificulta a adoção da ferramenta na indústria de software, pois sua configuração demanda conhecimento de otimização. Assim, o objetivo deste trabalho é apoiar usuários novatos ou inexperientes em algoritmos de busca na utilização da OPLA-Tool. Para isso, foram realizados experimentos para identificar as melhores combinações de parâmetros para configurar a OPLA-Tool em relação ao tipo de projeto de PLA. Os resultados obtidos serão úteis para guiar o desenvolvimento de uma funcionalidade cujo objetivo é permitir a configuração automática da OPLA-Tool de acordo com a entrada fornecida.

Introdução

Linha de produto de software (LPS) é uma abordagem de Engenharia de Software voltado à reutilização de artefatos de software para construção de famílias de produtos. Para o desenvolvimento de uma LPS é necessário projetar a Arquitetura da Linha de Produto (PLA), a qual é um dos principais artefatos de uma LPS (LINDEN, SCHMID e ROMMES, 2007). Porém, a obtenção da PLA não é uma tarefa trivial, pois fatores relacionados a diferentes propriedades arquiteturais devem ser atendidos, os quais muitas vezes são conflitantes, por exemplo, modularização de características, reusabilidade e extensibilidade da LPS (COLANZI et al., 2014). Dispor de apoio automatizado para projetar e avaliar uma PLA é de grande valia para o arquiteto de software.

Algoritmos genéticos multiobjetivo podem ajudar nesse sentido. Para fazer isso, Colanzi et al. (2014) propuseram a abordagem a MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*), que é uma abordagem que oferece um tratamento multi-objetivo para avaliar e otimizar projetos de PLA. Essa abordagem foi implementada na ferramenta OPLA-Tool (FREIRE et al., 2020) utilizando o algoritmo NSGA-II, um algoritmo genético multiobjetivo muito eficiente e

popular para resolver problemas de otimização. NSGA-II é um algoritmo multiobjetivo, pois permite a otimização simultânea de mais de um objetivo. Algoritmos genéticos, como o NSGA-II, são baseados na analogia com os processos de seleção natural e genética evolucionária (GOLDBERG, 1989), e incluem três tipos de operadores de busca: seleção, cruzamento e mutação. Como algoritmos genéticos são estocásticos, há um fator de aleatoriedade associado à aplicação dos operadores de busca.

No caso específico de otimização de projetos de PLA, os objetivos são constituídos por métricas de software relacionadas a propriedades arquiteturais que se pretende otimizar. A MOA4PLA possui operadores de mutação e de cruzamento específicos para a otimização de projeto de PLA, os quais foram implementados no NSGA-II. Para utilizar o NSGA-II é preciso configurar as probabilidades de aplicação dos operadores de mutação e de cruzamento, bem como qual o tamanho da população de indivíduos e o número de gerações do processo evolutivo. Em geral, esses valores variam de acordo com o problema em questão e com características dos indivíduos (projetos de PLA) fornecidos como entrada para o processo de otimização (GOLDBERG, 1989). Além disso, no contexto da MOA4PLA, que inclui operadores de mutação e de cruzamento específicos para o problema de projeto de PLA, é preciso selecionar quais operadores de busca (mutação e cruzamento) serão aplicados. A escolha da combinação destes operadores pode levar a diferentes resultados a depender das características do projeto de PLA fornecido como entrada pelo usuário para ser otimizada pela ferramenta.

Materiais e Métodos

No estudo de Freire et al. (2020) foram investigadas as melhores configurações da OPLA-Tool, ou seja, quais eram os melhores parâmetros a serem passados como entrada para a ferramenta. Tais parâmetros são: taxa de cruzamento, taxa de mutação, operadores de cruzamento e mutação, número de gerações, tamanho da população e número de avaliações de *fitness*. Esses parâmetros foram testados para as PLAs AGM e MM. Optou-se por utilizar as mesmas configurações para rodar os estudos experimentais envolvendo a PLA BET que é um projeto real e de grande porte, diferentemente da AGM e da MM que são LPSs acadêmicas.

O estudo foi realizado em duas baterias de testes, sendo que na primeira houveram duas *trials*. Estas baterias foram executadas nos servidores da UEM. Nas duas baterias de experimentos foram utilizadas as PLAs AGM, MM e BET. As 8 configurações usadas neste estudo estão na Tabela 1, sendo que na primeira bateria de testes foram testadas as configurações de 1 a 6 e na segunda bateria as configurações 7 e 8. Cada configuração foi executada 30 vezes, e o tempo de cada experimento variava de acordo com cada configuração. Para apoiar a realização desta atividade foi desenvolvido um algoritmo para normalizar as saídas de cada configuração, que eram um conjunto de soluções que a OPLA-Tool encontrou para a PLA BET. Cada solução é representada pelo seu valor *fitness* (um conjunto de três valores, sendo um para cada função objetivo) e por uma representação gráfica que consiste no projeto da PLA. Nesta atividade, a análise de resultados foi baseada em uma análise quantitativa, logo, apenas os valores de *fitness* foram utilizados para

fazer o comparativo entre as soluções de cada configuração. Esses valores foram normalizados e usados para calcular o *hypervolume*, que foi usado como indicador de qualidade na avaliação.

Foram comparadas as melhores configurações obtidas na primeira com a segunda bateria de testes, quando houve diferença estatística foi testado somente a config que apresentou um rendimento superior, já quando não houve diferença estatística entre as melhores configurações da bateria, ambas foram comparadas com a superior da outra bateria. Todos os testes estatísticos foram realizados entre duas configurações de cada vez, foram realizados os testes Shapiro Wilk para verificar a normalidade dos dados, Kruskal Wallis e Vargha Delaney para verificar a diferença estatística entre as configurações bem como o tamanho de efeito.

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

Config	Tamanho da População	Número 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
3	200	15	3000	FDC+CC	ALL	0.4	0.8
4	200	15	3000	FDC+MC	ALL	0.4	0.8
5	200	15	3000	FDC+CC	ALL	0.4	0.9
6	200	15	3000	FDC+MC	ALL	0.4	0.9
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

Para a primeira bateria de testes foi feito um comparativo entre as configurações que se destacavam para cada PLA. Para a MM2, que é a menor entre as PLAs analisadas, as melhores configurações foram a Config 4 e a Config 6, enquanto para a BET as melhores configurações foram a Config 1 e a Config 2. A partir disso foi formulado uma hipótese que para PLAs muito pequenas as Configs 4 e 6 sejam melhores (tamanho da população e de número de gerações baixos), enquanto para PLAs maiores as Configs 1 e 2 sejam melhores que as outras, pois envolvem populações grandes e maior número de gerações. Após a análise dos resultados da primeira bateria de testes, descritas anteriormente, observou-se que as melhores configurações para PLAs pequenas (AGM e MM) foram as Configs 1, 2 e 6, e para as PLAs grandes, as Configs 1 e 2. Apesar disso, foi observado que a utilização conjunta dos operadores de cruzamento Modular Crossover (MC) e Complementay Crossover (CC) pode ser benéfica em algumas situações (SILVA et al., 2020). Então decidiu-se por criar as configurações 7 e 8. A Config 7 foi proposta para atender as PLAs que são consideradas pequenas, ou seja, AGM e MM. Essa configuração possui a mesma configuração da Config 6, mas com os operadores de cruzamento MC e CC. A Config 8, por sua vez, foi proposta para ser executada para

PLAs grandes, ou seja, a BET. Ela possui a mesma configuração que a Config 1, porém com os operadores de cruzamento MC e CC.

Assim, na segunda bateria de testes foi possível observar que a Config 8 realmente atendeu as expectativas e se tornou a melhor configuração para a maior PLA no caso a BET, porém a Config 7 não demonstrou um comportamento superior às Configs 4 e 6, que era seu propósito, mostrando que para PLAs pequenas convém utilizar o operador Feature-driven Crossover (FdC). A Tabela 2 lista quais as melhores configurações por PLA ao final das duas baterias de testes.

Tabela 2: Melhores configurações para cada PLA

AGM1	AGM2	BET	MM1	MM2
Config1, Config2	Config2	Config8	Config2	Config4, Config6

Conclusões

A conclusão alcançada é que para propiciar a obtenção de melhores resultados, quanto maior o projeto de PLA, maior precisa ser a população e o número de gerações. Além disso, os operadores de cruzamento MC e CC foram benéficos para o projeto de maior porte, enquanto foi importante a presença do operador FdC para otimizar as PLAs pequenas.

Agradecimentos

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

Referências

COLANZI, T. E.; VERGILIO, S. R.; GIMENES, I. M. S. OIZUMI, W. N.. **A Search-Based Approach for Software Product Line Design**. In: The 18th International Software Product Line Conference (SPLC), 2014, Florence, Italy.

FREIRE, W.M., MASSAGO, M., ZAVADSKI, A.C., AMARAL, A.M., COLANZI, T. (2020). **OPLA-Tool v2.0: a Tool for Product Line Architecture Design Optimization**. Proceedings of the 34th Brazilian Symposium on Software Engineering.

GOLDBERG, D. E.. **Genetic Algorithms in search, optimization, and machine learning**. The University of Alabama, 1989.

LINDEN, F. van der; 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.

31º Encontro Anual de Iniciação Científica
11º Encontro Anual de Iniciação Científica Júnior



10 e 11 de novembro de
2022

SILVA, D. F., COLANZI, T. E., ASSUNÇÃO, W. K. G. **Enhancing Search-Based Product Line Design with Crossover Operators.** The Genetic and Evolutionary Computation Conference (GECCO) 2020, 2020, pp. 1250-1258.