

Calibração e experimentação de algoritmos genéticos multiobjetivos para otimização de projeto de arquitetura de linha de produto

Luiz Fernando Okada (PIBIC/CNPq/FA/Uem), Thelma Elita Colanzi (Orientador), e-mail: ra107247@uem.br

Universidade Estadual de Maringá / Departamento de Informática, PR.
Engenharia de software

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

Resumo

A abordagem de LPS (Linha de Produto de Software) tem sido usada por promover o reuso de software e assim reduzir o tempo de desenvolvimento de um produto. Um dos principais artefatos de uma LPS é a Arquitetura de Linha de Produto (PLA - *Product Line Architecture*). O projeto de PLA é uma atividade difícil que pode ser otimizada utilizando algoritmos de busca. A abordagem MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*) usa algoritmos de busca multiobjetivos para otimizar propriedades arquiteturais de um projeto inicial de uma PLA. Atualmente, o algoritmo utilizado nesta abordagem é o algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*), um algoritmo genético multiobjetivo que otimiza várias propriedades simultaneamente. É imprescindível estudar a configuração dos parâmetros do NSGA-II para obter os melhores resultados possíveis. Sendo assim, neste trabalho foram realizados experimentos para calibrar as taxas de aplicação dos operadores de cruzamento e mutação do NSGA-II para duas LPS. Como resultado foram descobertas as melhores taxas de cruzamento e mutação para operadores de busca do NSGA-II.

Introdução

Linha de Produto de Software (LPS) é uma abordagem que consiste no reuso de artefatos de software para uma família de produtos de um determinado domínio. Essa abordagem diminui o tempo de desenvolvimento do produto, pois promove a reutilização de artefatos já existentes. Um desses artefatos é a arquitetura de linha de produto (*PLA- Product Line Architecture*), otimizar este artefato pode ser uma tarefa difícil, contudo a abordagem MOA4PLA (*Multi-Objective Approach for Product Line Architecture*) [1] faz essa otimização utilizando algoritmos de busca multiobjetivos a fim de encontrar soluções com melhores propriedades arquiteturais, tais como modularização de características, extensibilidade da LPS, variabilidade e princípios básicos de um projeto inicial de uma PLA.

Atualmente, o algoritmo utilizado nesta abordagem é o algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) [2], um algoritmo genético multiobjetivo que otimiza várias propriedades simultaneamente. Algoritmos genéticos são baseados na evolução genética dos seres vivos e incluem operadores de busca de três tipos, incluindo seleção, cruzamento e mutação. A otimização do projeto de PLA ocorre por meio da aplicação aleatória desses operadores por várias gerações de populações de indivíduos formadas por alternativas de projeto de PLA obtidas aleatoriamente a partir de um projeto inicial.

Em um trabalho recente [3], dois novos operadores de cruzamento foram propostos para otimizar projetos de PLA no contexto da MOA4PLA. Nesse sentido, é imprescindível estudar a melhor combinação de configuração dos parâmetros do NSGA-II para obter os melhores resultados possíveis e também comparar os resultados obtidos pelos operadores de busca propostos com os operadores de mutação já existentes na MOA4PLA a fim de recomendar os operadores mais apropriados. Logo, o presente projeto teve como objetivo a realização de estudos experimentais com a finalidade de caracterizar os valores mais adequados para configurar o algoritmo NSGA-II para a otimização de diferentes projetos de PLA levando em consideração os operadores de busca propostos recentemente.

Materiais e métodos

Foram realizados três tipos de experimentos sendo eles: Exp_B - usando apenas operadores de mutação, Exp_FdC - utilizando operadores de mutação e o operador de cruzamento *FeatureDrivenCrossover* [3], e Exp_CC - utilizando operadores de mutação e o operador de cruzamento *ComplementaryCrossover* [3].

Para a realização dos experimentos foi utilizada a ferramenta OPLA-Tool [4], pois ela permite executar o NSGA-II com diferentes combinações de operadores de cruzamento e mutação configurando as taxas de aplicação dos operadores. Os experimentos foram realizados para duas PLAs: Arcade Game Maker (AGM) e Mobile Media (MM).

Em todos os experimentos foram realizadas 10 rodadas do algoritmo NSGA-II para cada PLA com os seguintes parâmetros: Tamanho da População: 100, Número de Avaliações de *Fitness*: 30000 (300 gerações) e as seguintes funções de fitness [5] (que devem ser minimizadas): ACLASS (acoplamento de classes), COE (coesão relacional) e FM (modularização de características de LPS). As taxas de mutação utilizadas variaram entre: 0.7, 0.8 e 0.9, enquanto as taxas de cruzamento foram: 0.0, 0.2, 0.4, 0.6 e 0.8.

A análise quantitativa dos resultados dos experimentos foi baseada no indicador de qualidade *hypervolume* e no teste estatístico Kruskal-Wallis para determinar o valor mais adequado para os parâmetros do NSGA-II. A comparação se deu entre os resultados de experimentos do mesmo tipo variando as combinações de taxas diferentes, ou seja, obtinham-se os

valores das três funções de *fitness* de cada solução gerada em cada rodada do NSGA-II, normalizavam-se os valores de *fitness* e a partir disso era calculado, o valor de *hypervolume* de cada uma das 10 rodadas do NSGA-II. A partir dos valores de *hypervolume* era feito um gráfico *boxplot* e então comparavam-se os gráficos para determinar a melhor combinação de taxas.

Resultados e Discussão

As Tabelas 1 e 2 mostram quais valores de parâmetros obtiveram melhores resultados em termos de *hypervolume* obtido nos experimentos realizados com as PLAs MM e AGM variando os valores das taxas de aplicação de cruzamento e mutação descritos anteriormente. Os valores destacados em negrito em cada linha são aqueles que cuja diferença foi atestada estatisticamente. Para os valores em que não houve diferença estatística, foi escolhida a maior mediana observada no *boxplot*.

Tabela 1: Melhores valores de parâmetros

Exp_FdC				Exp_CC			
	Taxa de Mutação	Melhor Taxa de Cruzamento	Melhor Taxa de Mutação		Taxa de Mutação	Melhor Taxa de Cruzamento	Melhor Taxa de Mutação
MM	0,7	0,4	0,9	MM	0,7	0,2	0,8
	0,8	0,4			0,8	0,2	
	0,9	0,4			0,9	0,2	
AGM	0,7	0,4	0,9	AGM	0,7	0,8	0,8
	0,8	0,4			0,8	0,4	
	0,9	0,8			0,9	0,4	

Tabela 2: Melhores valores de parâmetros para o Exp_B

PLA	Melhor Taxa de Mutação	Melhor Taxa de Mutação
MM	0,9	0,8
AGM	0,8	

Após a análise das tabelas selecionamos a taxa de cruzamento de 0,4 porque além de ter obtido maior *hypervolume* em 7 das 12 comparações, ela foi a única que teve diferença estatística em 5 comparações. Com relação à taxa de mutação, optamos pela taxa 0,8 que obteve o melhor *hypervolume* em 3 das 5 comparações, sendo o único valor que obteve diferença estatística em uma comparação entre experimentos do mesmo tipo.

Conclusões

Após a análise dos resultados, gráficos de *boxplots* e testes estatísticos realizados conclui-se que a melhor taxa de cruzamento é 0,4 enquanto a melhor taxa de mutação é de 0,8 para configurar os algoritmos NSGA-II para otimizar projetos de PLA utilizando os operadores de mutação e de cruzamento disponíveis atualmente na abordagem MOA4PLA.

Agradecimentos

Agradecemos ao CNPq pelo apoio financeiro e ao Diego F. da Silva e Aline S. Uchida pelo auxílio na realização do projeto.

Referências

- [1] COLANZI, T.E. **Uma abordagem de otimização multiobjetivo para projeto arquitetural de linha de produto de software**. 2014. 215 f. Tese (Doutorado) - Programa de Pós-Graduação em Informática, Universidade Federal do Paraná, Curitiba, 2014.
- [2] DEB, K.; PRATAP, A.; AGARWAL, S. and MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, 6(2):182 –197, Apr. 2002.
- [3] SILVA, D. F.; OKADA, L. F.; COLANZI, T. E.; ASSUNÇÃO, W. K.G. Enhancing Search-Based Product Line Design with Crossover Operators. In: GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, 20, 2020, Cancún. **Proceedings...** Disponível em < <https://dl.acm.org/doi/10.1145/3377930.3390215> >. Acesso em: 20 ago. 2020.
- [4] FÉDERLE, E.L.; et al. OPLA-Tool: a support tool for search-based product line architecture design. In 19th INTERNATIONAL SOFTWARE PRODUCT LINE CONFERENCE, 2015. **Proceedings...** Disponível em < <https://dl.acm.org/doi/10.1145/2791060.2791096> > . Acesso em: 20 ago. 2020.
- [5] VERDECIA, Y. D.; COLANZI, T.E.; VERGILIO, S. An Enhanced Evaluation Model for Search-based Product Line Architecture Design In: XX IBERO-AMERICAN CONFERENCE ON SOFTWARE ENGINEERING (CibSE 2017) - 39th International Conference on Software Engineering, 2017, **Proceedings...** Buenos Aires, 2017.