Search Algorithms for Regression Test Case Prioritization

Li, Zheng; Harman, Mark; Hierons, Robert M. (2007) "Search Algorithms for Regression Test Case Prioritization", IEEE Transactions on Software Engineering, vol. 33, no. 4, pp. 225-237, doi: 10.1109/TSE.2007.38

1. Fichamento de Conteúdo

Testes de regressão são executados em software para verificar se uma mudança introduziu algum erro no software. É um processo demorado. A priorização dos casos de teste visa permitir que falhas (testes que não passam) sejam descobertos o quanto antes. Li, Harman e Hierons (2007) defendem que não há solução ótima para o problema, podendo ele ser mapeado no problema da mochila, que é provadamente NP-difícil. Assim, eles investigam o desempenho de heurísticas na solução desse problema. Em particular, heurísticas baseadas em algoritmos busca (*Hill Climbing* e *Genetic Algorithms*) e algoritmos gulosos (*Greedy*, *Additional Greedy*, e *2-Optimal Greedy*). Experimentos são realizados com seis programas que variam de 374 a 11.148 linhas de código (LOC, *lines of codes*) e possuem suítes de teste de tamanho médio que varia de 230 a 4350 testes. Os resultados mostram que as heurísticas *Additional Greedy, 2-Optimal* e *Genetic Algorithm* são melhores que *Greedy Algorithm* puro. A explicação é que nesse tipo de problema há muitos ótimos locais, principalmente em grandes programas, de modo que, algoritmos baseados em busca global tendem a ser melhores do que algoritmos de busca local.

2. Fichamento Bibliográfico

- O algoritmo de Busca Gulosa (*Greedy Algorithm*) trabalha com o princípio do "próximo melhor", calcula um peso de cada teste em termos de maximizar a métrica de interesse (como a cobertura do código), e escolhe como próximo teste a ser executado, aquele que possui maior peso.
- O algoritmo de Busca Gulosa Adicional (*Additional Greedy Algorithm*) calcula o peso de cada teste buscando definir um próximo teste de modo a maximizar a parte dos testes que não foi atingida pelas decisões anteriores. Ou seja, está sempre buscando a parte adicional, não é métrica "geral".
- O *2-Optimal Algorithm* seleciona os próximos 2 testes que juntos maximizaram a métrica de interesse.
- Escalada ou subida da colina (*Hill Climbing*) é um algoritmo subótimo ou de ótimo local, seleciona o próximo teste baseado no 'vizinho' do teste atual que gera melhor resultado na métrica de interesse.
- Algoritmo Genético (*Genetic Algorithms*) define um cromossomo composto por todos os testes. Cada posição do cromossomo indica ordem em que respectivo teste será executado. A função de aptidão do indivíduo é calculada baseada no valor obtido para o indivíduo e na ordem do indivíduo na população. São realizadas mutações e cruzamentos. (No texto não está muito clara a condição de término)

3. Fichamento de Citações

- "Test case prioritization (...) orders test cases so that the test cases with highest priority, according to some criterion (a "fitness metric"), are executed first."
- "Given a function *f* that assesses the rate of achievement of code coverage, an efficient solution to the test case prioritization problem would provide an efficient solution to the knapsack problem, which is known to be NP-hard [7]. Thus, prioritization techniques for code coverage are necessarily heuristic [18]."
- "The following two research questions motivated this study: Q1. Which algorithm is most effective in solving the test case prioritization problem for regression testing? Q2. What factors could affect the efficiency of algorithms for the test case prioritization problem for regression testing?'
- "The results of the empirical study show that the Additional Greedy, 2-Optimal, and Genetic Algorithms always outperform the Greedy Algorithm. These results for the programs studied are shown to be statistically significant for all programs studied."
- "The fact that there are many local optima in the search space indicates that, for large test suites, the fitness landscapes are likely to be inherently multimodal. This result suggests that global search techniques could outperform local search techniques for regression testing prioritization problems, particularly for larger test suites."