Binary Multi objective PSO & GA for Adding New Features into an Existing Product Line

Abdullah Al Mamun, Faisal Al Naseef
g201403680, g200742170

Abstract

- In this work we propose a model for release problem considering product values, product integrity, and “AND” dependencies between features.

Introduction

- Maximizing the profits and customer requirements in Software Development Co.
- Next release problem

  What features in their customer's system would be developed first in the coming release.
- Research question

  Which and how new feature requests should be integrated into an existing Software Product Line.
- Objective

  To reach to a result that is suboptimal since our solution depends on meta-heuristics optimization techniques.
- Solution

  A Genetic Algorithm: NSGA-II

  A Particle Swarm Optimization Algorithm

Related works

- Next Release Problem

  The next release problem is firstly proposed by Bagnall et al [2].

  Zhang et al [3] generalized the original NRP problem, which is a single-objective optimization into a multi-objective NRP problem, defined mathematically in [4].

- Release Problem in Software Product Lines

  [5] They shown to deliver benefits in terms of effort reduction, quality improvement, and time-to market reduction.

  [1] Karimpour et al propose

  Product integrity: is defined as the degree to which the features are perceived as cohesive, and measured by nearest common predecessor (NCP) between two concrete features.

  Thum et al[6] considered specialization, refactoring, and generalization relationships between features.

  Ullah et al[7] proposed that structural impact should be minimized during updating a current system.

  GAFES (GA-based AI Approach to Optimized Feature Selection in SPLs). [8]

  NSGA-II algorithm [9] to solve them [10], [5], [7], and [3].

  We found that PSO algorithm is more efficient than GA in many works in petroleum engineering, mechanical engineering, and so on [9], [10], [11], and [12].

Problem Formulation

- Objective function

$$r_{value}(P) = \frac{1}{\sum_{j=1}^{p} w_j} \sum_{i=1}^{p} w_i V(p_i) \qquad (1)$$

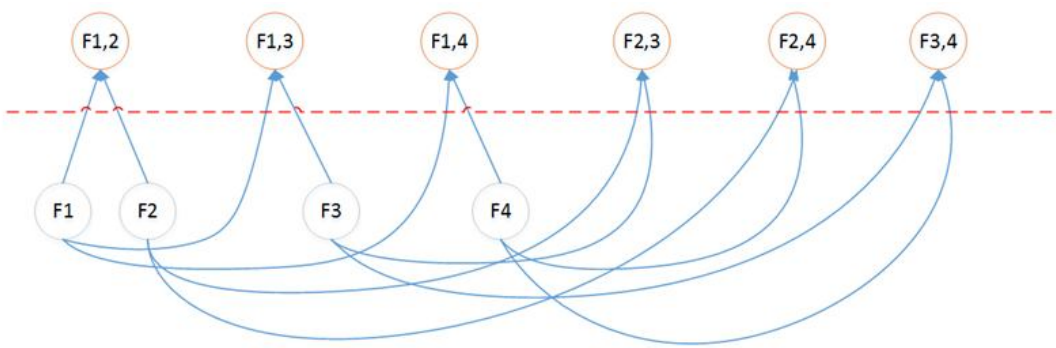
$$r_{intg}(P) = \frac{1}{\sum_{j=1}^{p} w_j} \sum_{i=1}^{p} w_i D(p_i) \qquad (2)$$

- P = set of all product variants for each P_i
  F = the set of all features and F_i
  w_i = the weight of product i.
  $V(P_i)$ return the product value and $D(P_i)$ the integrity value of product $P_{i.}$

Problem Formulation - 2

- Dependency

Dependent features $F_{1,2}$, $F_{1,3}$, $F_{1,4}$, $F_{2,3}$, $F_{2,4}$, and $F_{3,4}$



Independent features F1, F2, F3, AND F4

$$C_{(}a, b) = \frac{a!}{b!(a! - b!)} \tag{3}$$

a = Number of total independent features
b = Number of dependency combination
Example, C(4,2) = 6

Sample feature tree

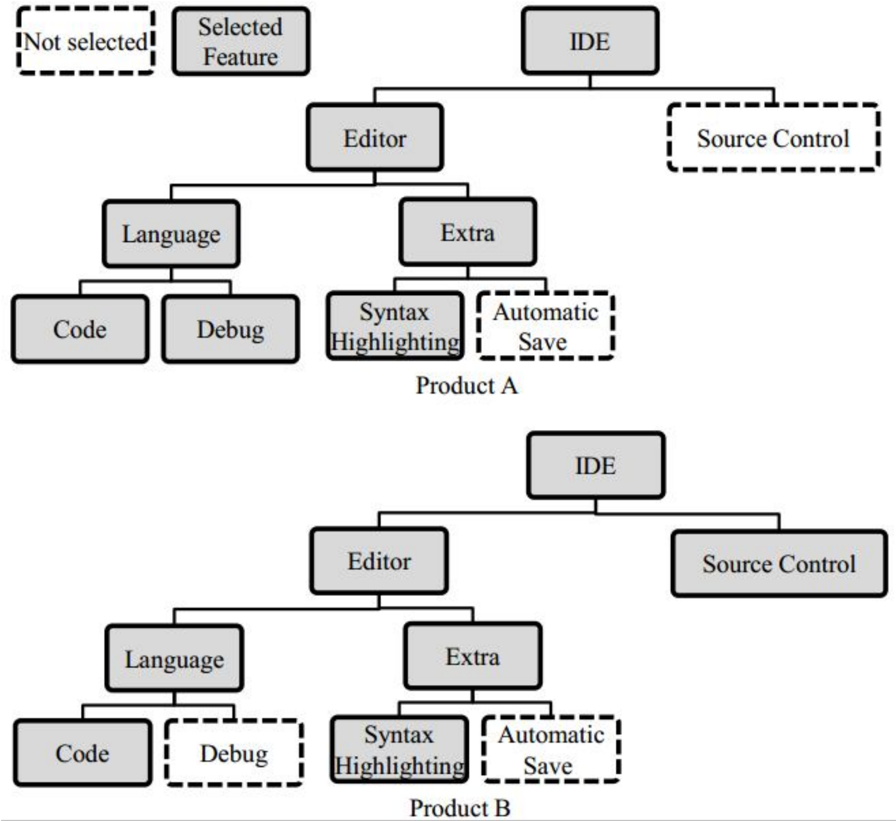


Fig. 2. Two products of a SPL. Product A has higher integrity compared to product B because of the distribution of the concrete features in sub trees

Methodology

- The Genetic Search Algorithm

  Hence, the problem is NP-hard for obtaining optimal feature selection as in [2]. In this work we applied NSGA-II algorithm for finding the best feature configuration given the feature model. For the implementation, we use jMetal library, which is a Java library containing a sort of meta-heuristic algorithms.

- The Particle Swarm Optimization (PSO) Algorithm
- We tried to follow an binary PSO implementation defined by [13]. The implementation was written in Java, using some components in jMetal library.

Methodology - 2

- <u>In feasible Chromosomes</u>
- Not all chromosomes generated in the evolution process are valid, despite their optimality. For this purpose, NSGA-II supports constraint values in the Pareto front calculation. We will be using this to discard feature models that do not represent feasible products.

- <u>Encoding</u>

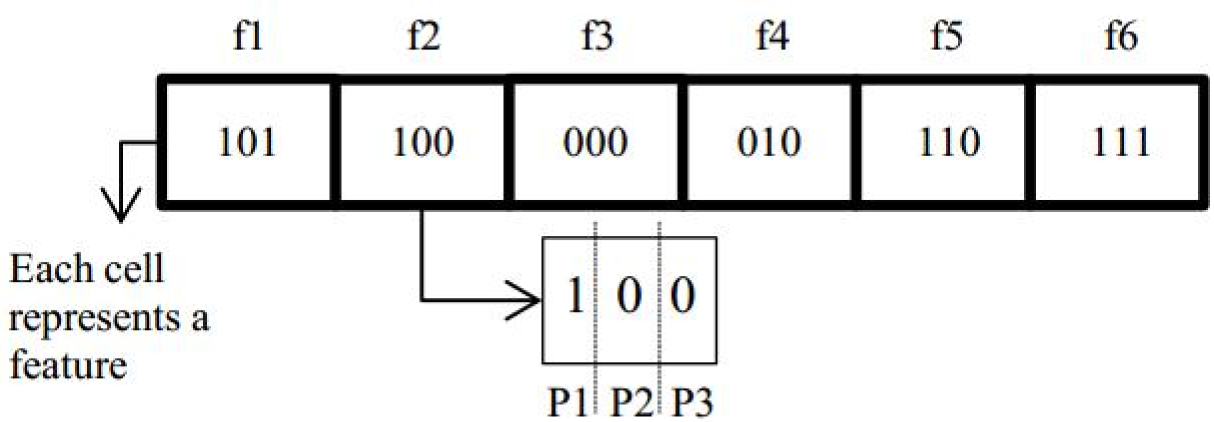


Fig. 3 Representing multiple products using a single chromosome.

Case study

Dataset

- Feature model: ELECTRONIC SHOPPING from SPLOT *(http://www.splot-research.org/)*
- Number of total features = 290
- Initially selected and implemented = 136
- Number of new feature = 30/154
- Product variants = 5
- Binary string size = 30 * 5 = 150 bit.

Results

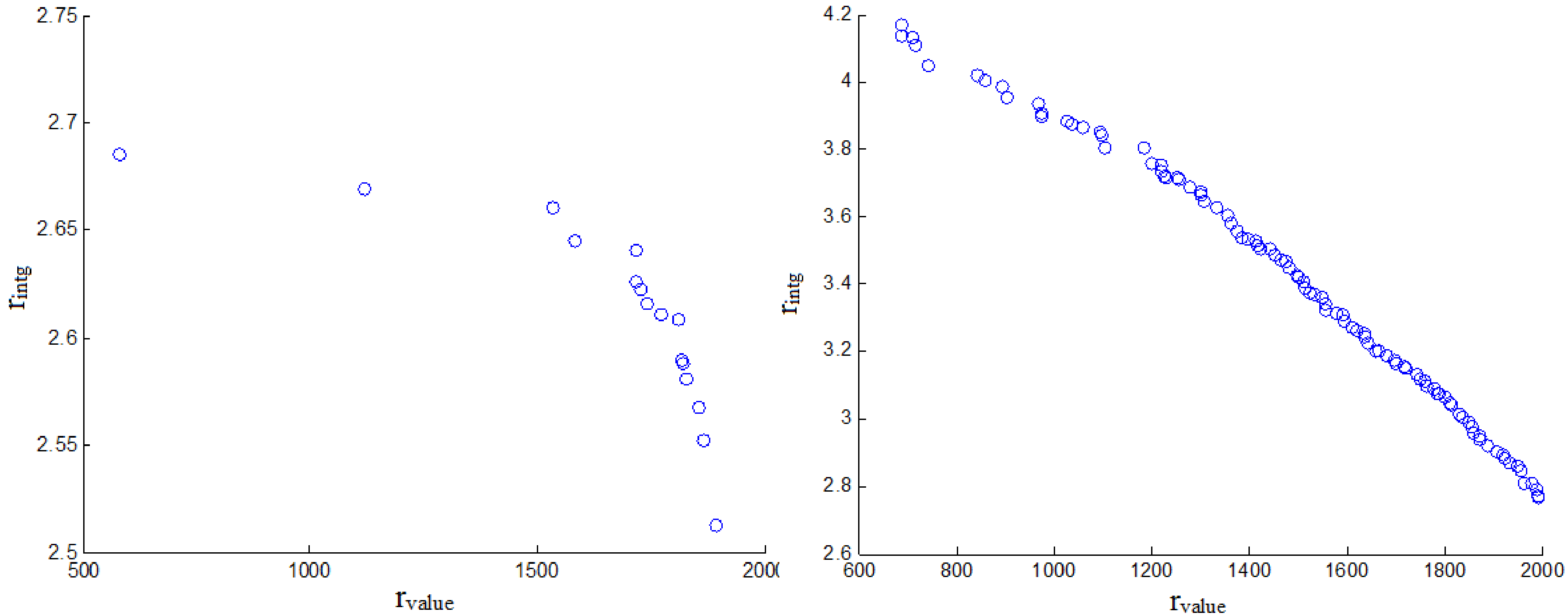


Fig.4 a) PSO b) NSGA-II

Generated output solution (Pareto front)

Result -2

File Edit Search View Encoding Language

FUN

```
-582.0 -2.685277686780063
-1121.0 -2.668940575178022
-1743.0 -2.6157864043822063
-1822.0 -2.58827086897626
-1775.0 -2.6107584801119565
-1855.0 -2.567437379128316
-1868.0 -2.552022143931015
-1893.0 -2.512634656462729
-1730.0 -2.6228060021790243
-1819.0 -2.589834449090943
-1810.0 -2.608444584534009
-1829.0 -2.5811394520244524
-1719.0 -2.640534129884261
-1585.0 -2.645392351121232
-1720.0 -2.6264344992698176
-1536.0 -2.660668971944917
```

File Edit Search View Encoding Languag

FUN FUN_NSGAII

```
-1993.0 -2.7653579925488767
-688.0 -4.171069065310149
-1993.0 -2.7653579925488767
-742.0 -4.050286156265301
-844.0 -4.019984139611596
-1186.0 -3.8035422264471
-1106.0 -3.8038237012112583
-968.0 -3.9351572375819153
-717.0 -4.108131284455123
-904.0 -3.9541849931130564
-895.0 -3.986942224527744
-1334.0 -3.6273991814223865
-1028.0 -3.882022692752801
-861.0 -4.007871485844632
-1200.0 -3.759482876389744
-1280.0 -3.6887180293615165
-1059.0 -3.8657545962061364
-977.0 -3.8972603906390115
-1355.0 -3.6021211636769714
-1308.0 -3.647489787909988
-1377.0 -3.558702825326998
-1645.0 -3.224518941324856
-1362.0 -3.581968767960377
-689.0 -4.136434460486639
-1963.0 -2.8088007563935764
-1095.0 -3.8492020492851164
-1483.0 -3.4514488751880057
-1956.0 -2.8475954345820975
-1831.0 -3.013447475900862
-1742.0 -3.130770779397136
```

Fig. 5 a) PSO b) NSGA-II

Generated R_{value} & R_{intg}

Binary output as selected optimal features

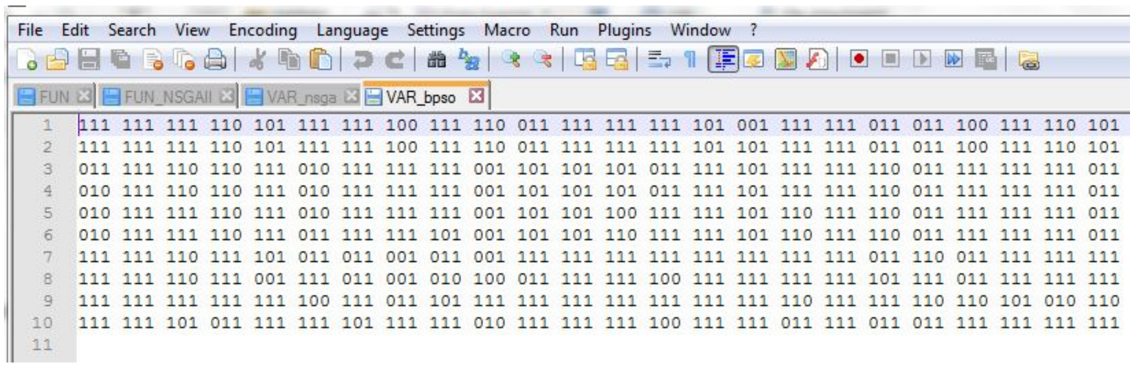


Fig. 6 PSO

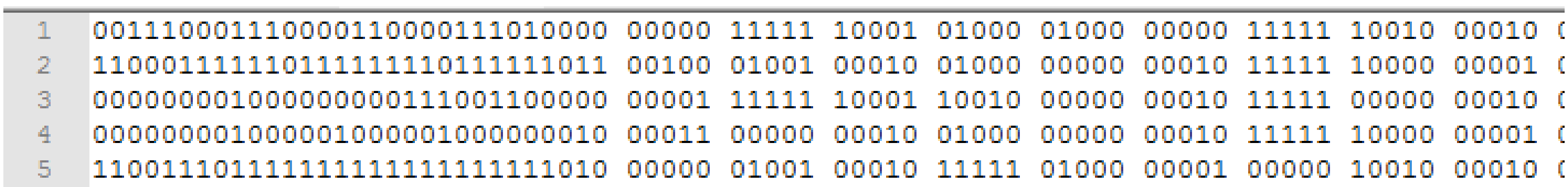


Fig. 6 NSGA-II

Conclusion

1. In this project, we presented two different techniques GA and PSO to solve the next release problem successfully.
2. We discussed features dependency and ANDing.
3. Successfully selected features those should or must include in next release based on Product value and product integrity.

Selected References

R. Karimpour and G. Ruhe, “Bi-criteria genetic search for adding new features into an existing product line,” in *Combining Modelling and Search-Based Software Engineering (CMSBSE), 2013 1st International Workshop on*. IEEE, 2013, pp. 34–38.

[2] A. J. Bagnall, V. J. Rayward-Smith, and I. M. Whittley, “The next release problem,” *Information and software technology*, vol. 43, no. 14, pp. 883–890, 2001.

[3] Y. Zhang, M. Harman, and S. A. Mansouri, “The multi-objective next release problem,” in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 1129–1137.

[4] X. Cai, O. Wei, and Z. Huang, “Evolutionary approaches for multiobjective next release problem,” *Computing and Informatics*, vol. 31, no. 4, pp. 847–875, 2012.

[5] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[6] S. Schulze, T. Thum, M. Kuhlemann, and G. Saake, “Variant-preserving¨ refactoring in feature-oriented software product lines,” in *Proceedings of the Sixth International Workshop on Variability Modeling of SoftwareIntensive Systems*. ACM, 2012, pp. 73–81.

[7] M. I. Ullah, G. Ruhe, and V. Garousi, “Decision support for moving from a single product to a product portfolio in evolving software systems,” *Journal of Systems and Software*, vol. 83, no. 12, pp. 2496–2512, 2010.

[8] J. Guo, J. White, G. Wang, J. Li, and Y. Wang, “A genetic algorithm for optimized feature selection with resource constraints in software product lines,” *Journal of Systems and Software*, vol. 84, no. 12, pp. 2208–2221, 2011.

[9] R. Rajendra and D. K. Pratihar, “Particle swarm optimization algorithm vs. genetic algorithm to solve multi-objective optimization problem in gait planning of biped robot,” in *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012*. Springer, 2012, pp. 563–570.

[10] E. Assareh, M. Behrang, M. Assari, and A. Ghanbarzadeh, “Application of pso (particle swarm optimization) and ga (genetic algorithm) techniques on demand estimation of oil in iran,” *Energy*, vol. 35, no. 12, pp. 5223–5229, 2010.

[11] A. C. Godinez, L. E. M. Espinosa, and E. M. Montes, “An experimental comparison of multiobjective algorithms: Nsga-ii and omopso,” in *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2010*. IEEE, 2010, pp. 28–33.

[12] L. D. Li, X. Yu, X. Li, and W. Guo, “A modified pso algorithm for constrained multi-objective optimization,” in *2009 Third International Conference on Network and System Security*. IEEE, 2009, pp. 462– 467.

[13] L. Wang, W. Ye, X. Fu, and M. I. Menhas, “A modified multi-objective binary particle swarm optimization algorithm,” in *Advances in Swarm Intelligence*. Springer, 2011, pp. 41–48.

[14]K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, “Feature-oriented domain analysis (foda) feasibility study,” DTIC Document, Tech. Rep., 1990.

Thank you

Q&A