

Exact Scalable Sensitivity Analysis for the Next Release Problem

Mark Harman* Jens Krinke* Inmaculada Medina-Bulo†
Francisco Palomo-Lozano† Jian Ren‡ Shin Yoo*

* CREST, University College London (UK)

† UCASE, Universidad de Cádiz (Spain)

‡ Beihang University (China)

Talk at the JISBD 2015
SISTEDES 2015



Introducing UCASE

UCASE

<http://ucase.uca.es>

- UCA Software Engineering Research Group
- Founded in 2011 with a broad focus in Software Engineering

Head

- Inmaculada Medina-Bulo (Inma)

Members

- 6 PhD
- 5 PhD candidates
- 7 / 11 members working on SBSE related areas

SBSE Related Research Lines

High Quality Mutant Generation

- Evolutionary Mutation Testing
- GAmera

<http://ucase.uca.es/gamera>

Testing of WS Compositions

- WS-BPEL Test-Case Generation
- Rodan

<http://ucase.uca.es/rodan>

Requirements Engineering

- Next Release Problem
- OATSAC

<http://ucase.uca.es/nrp>

An industrial example: Motorola

- Motorola identified 40 requirements for a new product
- A small number of requirements were identified as mandatory
 - Any release of the product must include at least these
 - A fix amount of budget is devoted to this task
- Decision-makers faced 35 **independent** requirements
 - Costs were estimated
 - Revenues were based in customer assessments

Which optional requirements to implement?

There are 2^{35} possible choices

- > 34000 million possibilities
- > 1000 million possibilities with half the total cost as budget

The Next Release Problem (NRP)

- Several variants defined by Bagnall et al. in 2001
- Simplest NRP is a classical knapsack problem
- Maximising revenues under budget constrains
 - Budget is “known” **a priori**
 - **Optimal solutions** exist and may not be unique

The Knapsack Problem and the NRP

- Knapsack problems first studied in the 1950s
- Basic variants of these problems are well understood
 - Most variants are **hard computational problems**
 - Simplest NRP is hard, but “weaker” than other hard problems
 - Different algorithms have been devised

Guarantees

- 1 Different guarantees for errors depending on the algorithm
 - No guarantee
 - Relative error
 - Absolute error
- 2 Most heuristic methods are “no guarantee” algorithms
 - Local search
 - Genetic algorithms
 - Simulated annealing

Facts

- 1 Guaranteeing absolute error for NRP is as hard as exactness
- 2 Relative error approximation algorithms exist

Sensitivity analysis

- 1 Estimation is not an exact science
- 2 Requirements are affected by estimation errors
 - Costs are invariably underestimated
 - Revenues are usually more predictable
- 3 Budgets are subjected to cuts (nowadays even more true)
- 4 Complex products go through different budgeting scenarios
- 5 Sensitivity analysis helps to identify the hot-spots

The Big Question

Q: How do you make sure that the observed variation stems from inherent sensitivity and not from the optimisation algorithm when you use an approximation algorithm?

A: You cannot!

Sensitivity analysis of the Motorola data

- 1 21 perturbations (from -50% to +50% with 5% steps)
- 2 Positive perturbations show the effect of underestimations

Challenges

Problem

- 1 Requirements are uncertain
- 2 Precise sensitivity analysis is key to decision making
- 3 Approximate solutions disable precise sensitivity analysis

Solution

Exact sensitivity analysis based on exact algorithms

Issues

- 1 Efficiency
- 2 Scalability
- 3 Interactions

Facts

- 1 Many algorithms have been devised in the last 50 years
- 2 Advanced algorithms are difficult to implement and test
- 3 For many algorithms space can be as critical as time
- 4 The curse of computational complexity on the NRP

No matter how good our exact algorithms are, they will always behave bad for an infinite number of instances, according to the current state of our knowledge (i.e., as long as $P \neq NP$)

Approach

- 1 Focus in “simple” algorithms and data structures
- 2 Algorithms have to be efficient just for the application domain

Bellman's equation for KP

$$z(n, B) = \begin{cases} 0 & \text{if } n = 1 \wedge c_1 > B \\ r_1 & \text{if } n = 1 \wedge c_1 \leq B \\ z(n-1, B) & \text{if } n > 1 \wedge c_n > B \\ \max\{z(n-1, B), z(n-1, B - c_n) + r_n\} & \text{if } n > 1 \wedge c_n \leq B \end{cases}$$

Nemhauser-Ullmann's algorithm (NU)

- 1 As values of z are computed, they are saved for later reuse
- 2 Solution is recovered from the values of z previously saved
- 3 In 2003, Beier and Vöcking proved very interesting properties
 - NU solves a random KP in **expected polynomial time**
 - This is so under **quite general conditions**
 - Strongly correlated instances are **provably harder** for NU

Correlation

- 1 Highly-correlated instances are reported hard in the literature
- 2 We control Pearson's correlation during instance generation

Scalability experiments

- 1 50 different datasets with 315 NRP random instances each
 - 15 problem sizes from 100 to 1500, with steps of 100
 - 21 correlation degrees from 0% to 100%, every 5%
- 2 Significant # of experiments: 15 750 instances solved
- 3 Experiments performed in just one core of a 1000 £ machine
 - Intel Core i7 2.67 GHz CPU with 12 GiB RAM
 - C++ on GNU/Linux
- 4 Results show that a polynomial model explains well our times
 - NU behaves **polynomially** in the number of requirements
 - Except when costs and revenues are **very highly correlated**

Assumptions

- 1 High correlations between cost and revenue are uncommon
- 2 This is particularly true in the case of optional requirements

One at a time

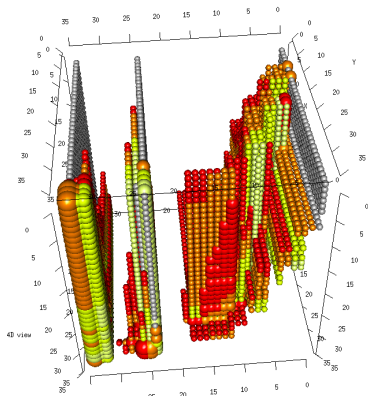
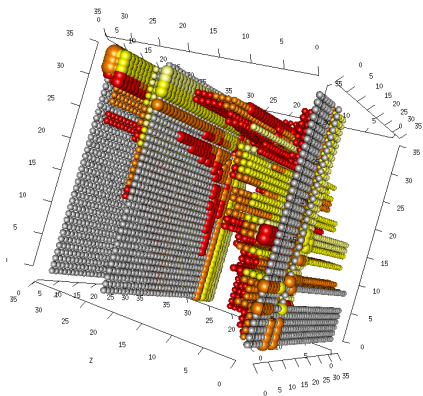
- Requirements are perturbed in isolation
- No interactions are taken into account

k at a time

- At most k simultaneous perturbations are considered
- Interactions are taken into account
- Very expensive

Second order interactions

- No interactions for a first analysis
- Interactions between pairs of requirements are then regarded



Conclusions

- 1 Exact algorithms enable precise sensitivity analysis (PSA)
 - Necessary to isolate estimation from approximation errors
 - Approximation errors may trick the decision-maker
- 2 Scalability is achievable, at least for reasonable cases
- 3 PSA is still expensive
 - 21 perturbations (from -50% to +50% with 5% steps)
 - $21 \cdot n$ instances for a single budget

However, we can complete a PSA without interactions for a project with 500 requirements in less than one day per budget scenario with just one core of a desktop computer

Parallelisation

- Different budgets can be analysed at the same time
- As well as different cost perturbations when a budget is fixed

SBSE for higher-order interactions

- SBSE could be used to look for higher-order interactions
- Especially to chase particularly insidious interactions
- NU would be used on demand to assess suspicious scenarios

Exact multi-objective NRP

- Maximising revenues while minimising costs
- Costs and revenues are **conflicting objectives**
- **Pareto-optimal solutions** instead of absolute, optimal solutions

Thank you for your attention

francisco.palomo@uca.es

Get more!



M. Harman, J. Krinke, I. Medina, F. Palomo, J. Ren, and S. Yoo.
Exact scalable sensitivity analysis for the Next Release Problem.
ACM Trans. Softw. Eng. Methodol., 23(2):19:1–31, 2014.

- Source code and experimental data freely available at:
 - <http://ucase.uca.es/nrp>
- Open access to the journal paper at:
 - <http://dl.acm.org/citation.cfm?id=2537853>
 - <http://dx.doi.org/10.1145/2537853>
 - <http://discovery.ucl.ac.uk/1428945>