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 Best of RESG Research 2014
British Computer Society

The Next Release Problem

1. Companies constantly release new products or versions
2. Requirements for a release can be of different kinds
   - Mandatory requirements vs. optional requirements
   - Decision-makers only control optional requirements
   - Release success depends on both kinds of requirements
3. Requirements are subject to constrains
   - Policy
   - Time-to-market
   - Dependencies
   - Budget
4. Requirements have expected estimations
   - Costs
   - Revenues
5. Which optional requirements to implement?

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
- Requirements can become mandatory by different reasons
  - Core product functionality
  - Dependence to other requirements
- Decision-makers faced 35 independent requirements
  - Costs were estimated
  - Revenues were based in customer assessments
- There are 2^{35} possible choices
  - $>$ 34000 million possibilities
  - $>$ 1000 million possibilities with half the total cost as budget

Formalisation

The Next Release Problem (NRP)

- Several variants defined by Bagnall et al.
- 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

Approximation algorithms

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

Greedy algorithms

Strategies

ICO	Increasing Cost Order
DRO	Decreasing Revenue Order
DRD	Decreasing Revenue Density

- DRO and DRD can suffer from unbounded error
- DRD with postprocessing is at worst 50% below the optimal

Advantages

1. Very fast and easy to implement
2. Do not impose artificial restrictions to their inputs
3. A natural basis for better algorithms, for example local search

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

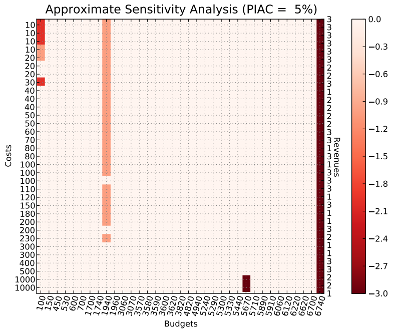
The Big Question

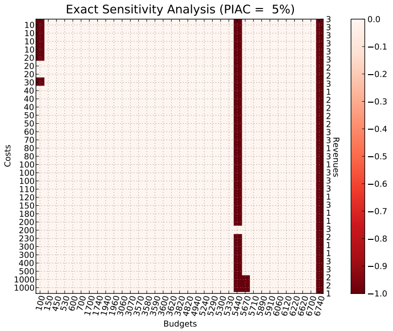
Q: How do you make sure that the observed variation stems from inherent sensitivity and not from the optimisation algorithm when you use an approximation algorithm?

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

Exact algorithms

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

Dynamic programming algorithms

Bellman's equation for KP

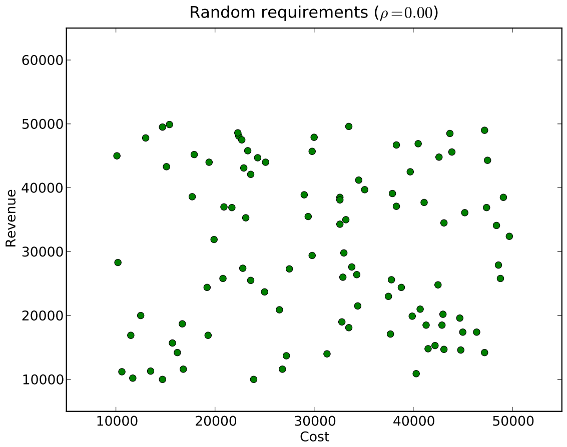
$$z(n,B) = \begin{cases} 0 & \text{if } n = 1 \wedge c_1 > B \\ r_1 & \text{if } n = 1 \wedge c_1 \leq B \\ z(n-1,B) & \text{if } n > 1 \wedge c_n > B \\ \max\{z(n-1,B), z(n-1,B-c_n)+r_n\} & \text{if } n > 1 \wedge c_n \leq B \end{cases}$$

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

Interactions

One at a time

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

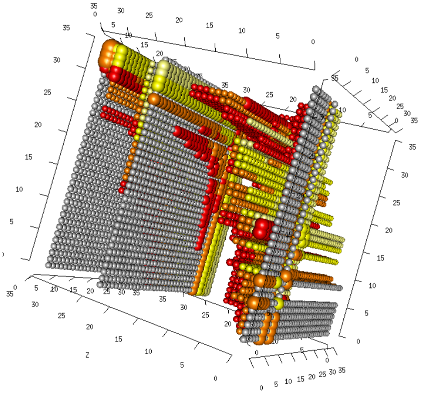
k at a time

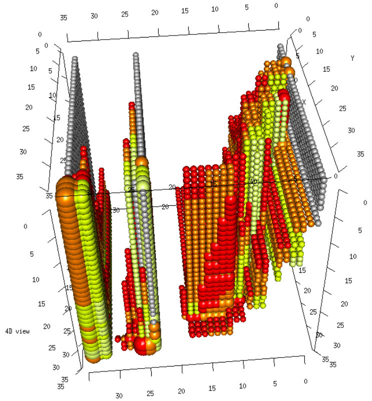
- At most k simultaneous perturbations are considered
- Interactions are taken into account
- Very expensive

Trade-off

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. However, PSA is still expensive
   - 21 perturbations (from -50% to +50% with 5% steps)
   - $21 \cdot n$ instances for a single budget

Still, we can complete a PSA without interactions of a project with 500 requirements for one budget proposal in less than one day

Future work

Multi-objective NRP

- Maximising revenues while minimising costs
- A Holy Grail
  - Costs and revenues are conflicting objectives
  - Budget may be known a priori or a posteriori
- Pareto-optimal solutions instead of absolute, optimal solutions
- Different approaches
  - Use multi-objective KP as a model to find the Pareto frontier
  - Produce the Pareto frontier as a byproduct of a KP algorithm

Parallelisation

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

Further future work

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

Better visual aids to decision making

- Better sensitivity visualisation for very large projects
- Interactive representations easy to manipulate

Thank you for your attention

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*