

Exact Analysis for Next Release Problem

Lingbo Li

CREST, Department of Computer Science
University College London
United Kingdom, WC1E 6BT
Email: lingbo.li@cs.ucl.ac.uk

Abstract—In software engineering, determining the set of requirements to implement in the next release is a critical foundation for the success of a project. Inappropriately including or excluding requirements may result in products that fail to satisfy stakeholders’ needs, and might cause loss of revenue. In the meantime, uncertainty is characterised by incomplete understanding. It is inevitable in the early phase of requirements engineering, and could lead to unsound requirement decisions. To ease the impact of uncertainty in the software development process, it is important to provide techniques that explicitly manage uncertainty in requirements analysis and optimisation.

This proposed research aims to provide a decision support framework for analysing uncertainty in requirements selection and optimisation. The proposed research involves three stages. Firstly, a simulation optimisation technique is introduced to model requirements uncertainty in requirements optimisation. Then, an exact technique is designed to eliminate the algorithmic uncertainty. Lastly, a probabilistic uncertainty analysis is applied to help the decision maker to understand requirement uncertainty propagation and the characteristics of requirements in requirements selection process.

I. INTRODUCTION

The term *software requirement* is defined as “the property which must be exhibited in order to solve some problem in the real world” [1]. Essentially, a software requirement is typically a complex combination of sub requirements to satisfy different stakeholders and deployment environments [2].

Determining an appropriate subset of requirements to be delivered in the next release of a software system is a critical aspect in software engineering, especially under limited resources. In 1996, Karlsson innovated *Analytical Hierarchy Process* for supporting software requirements selection and prioritisation [3]. It was further formulated as Next Release Problem (NRP) by Bagnall et al. [4] in 2001. The NRP models stakeholders’ objectives quantitatively, and employs optimisation techniques (i.e., meta-heuristic algorithms, dynamic programming) to explore and select a subset of requirements that is both feasible and well-suited to stakeholders’ requirements. The problem is similar to the Knapsack Problem, which is known to be *NP-hard* [5]. The search space of this problem increases exponentially with the number of requirements.

Unfortunately, uncertainty is also an inherent characteristic of the software engineering process [6]. The essence of uncertainty is the lack of complete knowledge at the time a decision must be made [7]. In requirements engineering, requirements are uncertain. Requirements are often incomplete, vague and subject to change. The requirements uncertainties include

uncertainty about the development resource availability, the impact of dynamic and frequent changes in whole software development life cycle, and the accuracy of the software project estimation. The requirements of a new system are uncertain if the users have not started to use it [8]. Decision makers have to make decisions under such uncertainties. Underestimated or ignored uncertainties may bring risks into software projects, and might even result in project failure [8].

Over the last decade, various NRP techniques have been developed in a context where the input requirements’ attributes are concerned with point-based estimations, which are estimated by human requirements engineers [4], [9], [10]. In those studies, the attributes of requirements and stakeholders are quantified as explicit value, and requirements uncertainty is either underestimated or completely overlooked [11]. For example, given a set of quantified requirements, although those point-based estimation approaches can provide optimal solutions in terms of expected cost and revenue, they fail to offer an assessment of the confidence of such results. Thus, they may mislead the decision making and amplify the consequences of risks.

In order to manage the uncertainty in requirements selection and optimisation, two approaches have been conducted in previous work. The first approach is sensitivity analysis, which is an uncertainty handling method. The purpose of performing sensitivity analysis is to evaluate the robustness of the outputs of a model in the presence of uncertainty, and to achieve uncertainty reduction through identifying sensitive model inputs. Usually, sensitivity analysis applies changes in the input of the system, and studies what effect this produces on the output. The second approach is robust optimisation [12]. Robust optimisation is distinctly different from sensitivity analysis. Robust optimisation formulates the optimisation problem as one in which solutions that have a priori ensured robustness are sought against prescribed uncertainty [13]. Straightforwardly, robust optimisation explores the solution space and takes uncertainty into account simultaneously.

On the other hand, to deal with uncertainty, it is important to know that all uncertainty derives from the problem itself and not from the algorithm used to tackle it. Nevertheless, the previous NRP uncertainty management studies adopted non-deterministic approaches in their framework [14], [15]. These approaches do not guarantee to find the optimal solutions, therefore, may return solutions that are worse than optimal. In other words, there is information loss in the solution,

and additional uncertainty is thereby introduced. While such information loss is acceptable in general, for the specific problem of handling uncertainty we face here, it is important to ensure that the resulting uncertainty is from the problem itself. This motivates the usage of exact algorithm for uncertainty handling. However, the designed exact algorithm is problem-dependent and it is very time-consuming [16]. The execution time may increase exponentially respect to the number of input parameters and the dimensions (number of objectives) of the problem.

The principal goal of this research is to provide a framework to manage uncertainty exactly and efficiently in requirements selection and optimisation. The detailed aims and objectives of this thesis are as follows:

- 1) Investigating the feasibility of applying probabilistic robust (simulate-based) optimisation for managing uncertainty in NRP.
- 2) Proposing an exact NRP optimisation technique to guarantee to find the optimal solution in a finite amount of time.
- 3) Providing a probabilistic uncertainty analysis and quantitative analysis framework to help the decision makers to study requirements uncertainty propagation in requirements optimisation process, and interpret the produced results.

II. EXISTING WORK

This section presents a preliminary literature review to establish a foundation for undertaking this research. The research of this work intersects two main research fields, including the field of requirements selection and optimisation, and uncertainty handling.

A. Requirement selection and optimisation

Requirements may have different values to stakeholders, and require different efforts to implement. In an ideal world, all requirements would be simply selected and implemented in one release, thereby yielding maximal functionality and value to stakeholders. However, in practice, resource constraints need to be taken into consideration. Given a limited budgetary resource, it is impossible to implement all requirements in one release. Requirement engineers have to make decision to determine the priority of requirement and decide whether this requirement should be implemented in the next release of the system or not, meanwhile, the outcome of the software system in the next release can be maximised [1].

Essentially, the requirements selection and prioritisation activity are concerned with selecting a subset of requirements to implement to meet the demands of stakeholders and maximise the value of delivered software product, at the same time, scale to ensure that there are sufficient resources to undertake the development [2]. Such requirements optimisation problem is recognised as a complex combination problem.

In the literature, various techniques have been developed to address requirements optimisation problem. Technically, the requirements optimisation techniques can be categorised

into priority-based requirements optimisation, search-based requirements optimisation, and exact requirements optimisation [17].

Priority-based requirements optimisation is an intuitive approach in which the requirements are sorted from ‘best’ to ‘worst’ based on its characteristics or the interests of the stakeholders. The ranking of requirements implicitly indicates the priority of requirements. Thereupon then, the developers can choose the requirements with the highest priority to implement, accordingly, achieve the earliest satisfaction. Some priority-based approaches have been proposed in the literature to support requirements selection and optimisation, such as AHP [3], [18], Quality Function Deployment (QFD) [3], and StakeSource2.0 [19], a web-based requirements prioritisation tool that uses ‘crowdsourcing’ approach. However, the priority-based approaches are restricted by: 1) the prerequisite of that stakeholders should be familiar with at least one of the prioritisation methods; 2) the required effort is most likely to be overwhelming.

According to Search-based Software Engineering (SBSE) which was coined by Harman and Jones in 2001 [5], the complex, multi-objective, and highly constrained software engineering problems can be formulated as search-based combination optimisation problem that can be tackled with the search algorithm. To convert a software engineering problem into a computational search problem, a fitness function is needed to measure the quality of candidate software engineering problem solutions. Naturally, the requirements optimisation problem is requirements combination problem, and can be viewed as an application area for SBSE [4], [20], [21].

Bagnall et al. [4] proposed the term *Next Release Problem* (NRP), and attempted to formulate requirements selection and optimisation as a combination-based requirements release planning problem. The NRP model assumes that there is a set of stakeholders and their features in the next release of a software system. The set of stakeholders is denoted by $C = \{c_1, \dots, c_m\}$ and the set of possible requirements is denoted by $R = \{r_1, \dots, r_n\}$, where m is the number of stakeholders, and n is the number of features.

During the software development, some resources (e.g., human resources and facility resources) need to be allocated to satisfy each requirement. NRP uses cost to measure the amount of resource needed to fulfil the requirement as given by $Cost = \{cost_1, \dots, cost_n\}$.

There is a weight vector which reflects the degree of importance of each stakeholder for the company. The relative weight vector related to each stakeholder c ($1 \leq j \leq m$) is denoted as $Weight = \{w_1, \dots, w_m\}$, subject to: $w_j \in [0, 1]$, and $\sum_{j=1}^m w_j = 1$.

The importance of each requirement for each stakeholder is assumed different. Given a stakeholder, the level of satisfaction of this stakeholder is based on the requirements that are satisfied in the evolved suggestion for the next release of the software system. Based on this assumption, each requirement r_i ($1 \leq i \leq n$) is assigned a *value* (r_i, c_j) by each stakeholder c_j ($1 \leq j \leq m$). The overall revenue of a given

requirement r_i ($1 \leq i \leq n$) for the company is denoted by $Revenue_i = \sum_{j=1}^m (w_j \cdot value(r_i, c_j))$.

In NRP, the requirements selection and optimisation solution is presented as a decision vector $\vec{x} = \{x_1, \dots, x_n\} \in \{0, 1\}$ to determine the requirements that are to be selected in the next release. In this vector, x_i is 1 if requirement i is selected and 0 otherwise.

On the basis of NRP, several NRP formulations or problem statements have been investigated recently. For example, dealing with multiple criteria [10], [22], [23], and evaluating the performance of meta- and hyper- heuristics [24].

Though search-based approaches might provide an approximate solution in a reasonable time and scale well, in special cases, decision makers need exact approach to guarantee the exactness of the results. In this research, using approximate approach to analysis and manage the uncertainty in NRP might introduce uncertainty from the nature of the algorithms used to optimise. Therefore, our proposed approach is focusing on exact optimisation approach.

Exact optimisation method is the optimisation method that can guarantee to find all optimal solutions. In principle, the optimality of generated solution can be proofed mathematically. Therefore, exact optimisation is also termed as mathematical optimisation. However, the exact optimisation approach is impractical usually. The effort of solving an optimisation problem by exact optimisation grows polynomially with the problem size. For example, to solve a problem by brute force approach, the execution time increases exponentially respect to the dimensions of the problem.

The idea of applying exact optimisation approach on requirements selection and optimisation is similar with search-based requirements optimisation. The only difference is that, instead of using search-based optimisation algorithm, the search-based requirements selection and optimisation problem is tracked with exact optimisation algorithm. There are three main categories exact optimisations found in the literature. They are Integer linear programming [25], dynamic programming [26], and exhaustive search [27].

B. Uncertainty Handling in Requirements Selection and Optimisation

Uncertainty is ubiquitous and accompanies all events in the real world. It covers all fields of scientific studies, and is inevitable in many aspects of decision making [28]. The essence of uncertainty is the lack of complete knowledge at the time a decision must be made [7]. Uncertainty arises from different sources in various forms, and complicates and affects decision making [28]. Even worse, it may arise the consequences of events to be risky. For example, uncertainty about some kinds of undesirable events such as machine failures, disasters occur, or system fails in delivery on time.

Though it is hardly likely that uncertainty could be eliminated completely, it is worthwhile to identify and handle with uncertainty to avoid unfavourable hazards [29]. To provide a confidence final decision, there are two straightforward approaches to cope with uncertainty so far [30], [31]. One

of the approaches is to analyse uncertainty as a post-analysis method [32]. Another one is robust optimisation, an approach that includes modelling and optimising the systems while taking uncertainty into account [12], [33].

According to the literature and practice, the former approach consists of two inter-related approaches to analyse uncertainty [34], [35]. These are sensitivity analysis and uncertainty analysis. The sensitivity analysis is performed in order to identify variations in results obtained from original and perturbed model input values [34], [36]. It is the study of how the individual uncertainty of model input contributes to the overall uncertainty of a model output [14], [26]. This offers the knowledge about which one of input drive the majority of the variance in the output. Meanwhile, the uncertainty analysis attempts to explain the possible outcomes, together with their associated possibility of occurrence [35]. The uncertainty analysis thus measures the overall uncertainty of the conclusions of the model [37]. Comparing with sensitivity analysis, uncertainty analysis concentrates on uncertainty quantification and propagation of uncertainty.

Though analysing uncertainty can evaluate how sensitive the solutions are to possible estimation uncertainties, it cannot offer robust solutions by itself, based on decision makers' degree of risk aversion. Hans-Georg and Sendhoff [12] suggest investigating uncertainties *during* the process of optimisation rather than using post-analysis. This approach was termed as *robust optimisation* [38]. The robust optimisation is regarded as the approach that searching and optimising the solutions that are immune on production tolerances, parameter drifts, and model sensitivities.

The applications and studies of robust optimisation can be widely found in other non software engineering research literature [39], [40] but are seldom found in the requirements engineering research. In requirements engineering, to the best of our knowledge, there are only three studies applying robust optimisation on requirements optimisation area.

In 2011, Heaven and Letier first proposed a search-based optimisation framework, which integrated with stochastic simulation, for guiding the choice of system design solutions on high level goals in quantitative Goal Models [41]. Paixão and Souza were the first authors to introduce a robust optimisation framework to the NRP problem in 2013 [15]. They used the interval to model the uncertainties of requirements implementation cost, and defined a small population of scenarios to represent the uncertainty of requirement value. Their robust NRP model tries to maximise the overall release solution value for all possible scenarios, while minimising the implementation cost of release solution for the worst case. Thereby, the outcome of their approach is a conservative robust solution, which can avoid the impact of uncertainty in the worst scenario. In 2014, Leiter et al. applied statistical decision theory to illustrate the *expected information value* of model parameters based on [41] to offer further decision suggestions [27]. To overcome the limitations of approximate meta-heuristic algorithms, the exhaustive search was adapted to explore the full solution space. The statistical *expected*

information value was computed to explain the ‘robustness’ of a solution, and then treated as an/the objective for maximising.

III. RESEARCH QUESTIONS AND CHALLENGES

To evaluate the proposed research, we derived four research questions, each of which builds on its predecessor to develop the evidence for the validity, performance, effectiveness and insights gained from proposed framework to manage uncertainty in requirements selection and optimisation.

RQ1: How does the proposed probabilistic robust (simulation-based) optimisation model perform compared to non-uncertainty-aware optimisation model?

RQ2: What is the effectiveness of proposed exact NRP solver for eliminating algorithmic uncertainty compared to NSGA-II?

RQ3: What is the impact of the requirements uncertainty?

RQ4: Is there any pattern between the requirements characteristics and requirements inclusions? If so, what kind of pattern can be observed?

The first question is a preliminary ‘sanity check’ question. If the proposed model cannot lead the uncertainty-aware optimisation technique to outperform non-uncertainty-aware optimisation technique convincingly w.r.t robustness, then it is clearly that model formulation is not correct. After validating the correctness of proposed NRP mode, the second research question will investigate how much difference can be observed between the NRP solutions found by NSGA-II and proposed exact NRP solver. This research question is a foundation for applying our exact NRP solver on managing uncertainty in NRP. We would compare the solutions found by NSGA-II with the benchmarks which are found by proposed exact NRP solver. The differences between NSGA-II solutions and benchmarks reveal additional (unnecessary & unhelpful) uncertainty introduced by NSGA-II. Some metrics (e.g., product failures probability reduction, or lost revenue reduction) would be proposed to measure these differences to enable better understanding for decision makers. In turn, this research question validates the effectiveness of proposed NRP solver. The third question is concerned with scrutinising the impact of uncertainty that came from requirement itself. It can be expressed in a quantified manner as to how much expected robustness can be obtained when a decision moves from an optimal-yet-risky solution to a robust-yet-suboptimal one under the same budget. The last question investigates the possible insight of the requirement characteristics, which may help decision makers to concentrate on the most interesting property of requirements. The same metrics used in research question two can also be applied for answering this question.

In order to achieve the goals of this research, three challenges have been identified. The first challenge is concerned with the generality of the model. Since different team/organisation may use different measurements of uncertainty, the proposed uncertainty formulation of problem solution may be become invalid. To solve this challenge, we intend to use simulation technique to formulate uncertainty,

rather than deriving uncertainty formulation mathematically. The second challenge is intrinsically associated with exact NRP solver designing. It is widely known that exact optimisation approach is problem specified. It tailored to perform on a specific problem. Moreover, the effort of solving an optimisation problem by exact optimisation grows polynomially with the problem size. Especially, dealing with multi-objective makes the performance of exact optimisation worse. To design a scalable and efficient optimisation algorithm is mathematically difficult. One way to solve this challenge is designing a requirements interaction schema to extend the existing exact NRP solvers [25], [26]. As the proposal intend to help decision makers to understand the impact of requirements uncertainty, the last challenge is to establish a decision support tool to interpret the results produced. In this case, the proposed framework must be able to provide more details about the nature of requirement properties and visualise them, which may inspire decision makers to prioritise the requirements for further evaluation and inclusion.

IV. RESEARCH METHOD AND PROGRESS

Conducting a systematic review to capture the state-of-the-art in the area related to this proposed research is the research activity throughout the whole research. It would establish the theoretical foundation for the research undertaken in this proposal. The rest steps of the research will be carried out based on the knowledge obtained from the literature review. As mentioned before, each of our proposed research questions builds on its preceding research question. Consequently, there are four research steps would be carried out to answer the research questions sequentially.

To answer the RQ1, we were looking for a general uncertainty model to formulate requirements uncertainty in NRP. The literature review convinced us that the numerical uncertainty modelling techniques [42] rely on the strong mathematical assumption to measure the uncertainty, which needs explicit the complex mathematical information. By contrast, as described in Section II-B, simulation optimisation only needs the probabilistic interpretation of uncertainty and uses the simulation techniques as the statistical inference tools to estimate the “true” values of the function based on the law of large numbers. It is reconsigned as an ideal and universal tool, which uses random iterates combining with optimisation approaches, to solve stochastic uncertain problems. In other words, decision makers can easily adapt proposed model to handle different kinds of uncertainty when they have a probabilistic interpretation about the uncertainty. We have conceptually built a simulation-based NRP model to validate the correctness of our proposed model. The result of this step has been published in [43].

The second research step is to develop an effective exact NRP solver to deal with algorithmic uncertainty. Much of the previous work on NRP is concerned with the application of search based techniques. While this is acceptable in general, for the specific problem of handling uncertainty we face here, it is crucial for the decision maker to know that all uncertainty

derives from the problem itself and not from the algorithm used to tackle it. To attack uncertainty in NRP, the most intuitive way is applying exhaustive search. However, it is inherently expensive and may not scale sufficiently to be more generally applicable. For a NRP model which consists of n requirements, there are 2^n solutions in the objective space. In order to achieve the effectiveness, there are two classes optimisation approaches. The first approach is linear programming, while another one is dynamic programming. The former one has some issues to be solved, such as scalability (cannot deal with more than three objectives simultaneously). The latter one is concerned about the issue that fails to take care of the interactions between the problem elements. Meanwhile, our research should take into account multiple objectives and the problem constraints. We would empirically evaluate the effectiveness of the designed optimisation approaches with the state-of-the-art search based techniques over different problem sizes and different degrees of requirement interaction. We would develop some metrics (e.g., project failure probability reduction, or lost revenue reduction) and use these metrics to inspect the differences between the solutions generated by proposed exact NRP solver and the search technique to promote the motivation of designing exact approach.

The next step is to examine the usefulness of proposed framework to deal with requirements uncertainty. For this purpose, a controlled empirical study which covers extreme cases and ‘in-between’ case will be conducted. The proposed uncertainty-aware approach will be performed on NRP problem together with the non-uncertainty-aware approach to obtain robust-yet-suboptimal solutions and optimal-yet-risky solutions. Some metrics would be proposed to measure the impact of requirements uncertainty. The results of comparison are statistically analysed to test, understand, and interpret the usefulness of our approach.

The final research step is to investigate the possible insights of requirements characteristics. In this stage, a set of decision support tools and visualisation tools are developed to help requirements engineer better concentrate on detail property of requirements. The requirement properties may include the requirement sensitivity, the impact of requirement interaction, and the requirement similarity.

V. CONTRIBUTIONS

This proposed research introduces a decision support framework for analysing the uncertainty in requirements selection and optimisation process. It may contribute to the requirements engineering by:

- 1) Introducing a probabilistic robust (simulation-based) optimisation model to probabilistically evaluate requirements uncertainty, and formulate the simulated requirements uncertainty as one of the objectives for optimisation. This model may enable a decision maker to analyse and optimise requirements and takes uncertainty into account simultaneously.
- 2) Designing an exact NRP optimisation solver that can guarantee to find the optimal solutions and eliminate

algorithmic uncertainty. By virtue of this exact approach, decision makers can ensure that the variations between the fragile, but optimal, results and the conservative non-optimal results derive from the inherent uncertainties of the requirements, thus the stochastic nature of the approximate algorithms can be excluded.

- 3) Providing a decision support framework that allows decision makers to study requirements uncertainty propagation in the requirements optimisation process probabilistically, and interpret the results produced.

VI. CONCLUSION

In software engineering, requirements analysis and decision analysis are the critical foundations of the success of a software project, since the uncertainty is essentially inevitable in early requirements engineering. The requirements engineering community has demonstrated the success of quantitative multi-objective decision techniques and search-based approaches to produce optimal solutions to decision makers in the past decades [9], [10]. Decision makers are informed of possible trade-offs among conflicting objectives by visualising the Pareto optimal solutions generated by these quantitative methods. However, little work has been done to model design time uncertainties, interpret the consequences of those uncertainties, and support decision makers in analysing the inherent characteristics of model parameters [14], [26], [27], [41]. The decisions have to be made under incomplete knowledge about software project.

The goal of the proposed research is to better support requirements engineers in understanding and analysing the inherent characteristics of requirements uncertainty. To achieve this goal, we propose a simulation based NRP to model requirements uncertainty, and introduce an exact analysis approach to support requirements selection and optimisation in the presence of uncertainty, as well as designing a decision support framework to help decision makers to understand the impact of requirement uncertainty.

PUBLICATIONS RELATED TO THE RESEARCH

- 1) L. Li, M. Harman, E. Letier, and Y. Zhang, *Robust next release problem: handling uncertainty during optimization*, In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14). ACM, New York, NY, USA, 1247-1254.

ACKNOWLEDGEMENT

The work presented herein was supported by the China Scholarship Council (CSC) and the UK EPSRC project EP/J017515 (DAASE). The author is a Ph.D student under the joint supervision of Prof Mark Harman, Dr Emmanuel Letier, and Dr Yuanyuan Zhang.

REFERENCES

- [1] A. Abran, P. Bourque, R. Dupuis, and J. W. Moore, Eds., *Guide to the Software Engineering Body of Knowledge - SWEBOK*. Piscataway, NJ, USA: IEEE Press, 2001.
- [2] I. F. Alexander and L. Beus-Dukic, *Discovering requirements: how to specify products and services*. John Wiley & Sons, 2009.

- [3] J. Karlsson, "Software requirements prioritizing," in *Requirements Engineering, 1996., Proceedings of the Second International Conference on*, Apr 1996, pp. 110–116.
- [4] A. J. Bagnall, V. J. Rayward-Smith, and I. M. Whittle, "The next release problem," *Information and Software Technology*, vol. 43, no. 14, pp. 883–890, 2001.
- [5] M. Harman and B. F. Jones, "Search-based software engineering," *Information and Software Technology*, vol. 43, no. 14, pp. 833–839, 2001.
- [6] H. Ziv, D. Richardson, and R. Klösch, "The uncertainty principle in software engineering," in *Proceedings of the 19th International Conference on Software Engineering (ICSE'97)*, 1997.
- [7] J. Helton, J. Johnson, C. Sallaberry, and C. Storlie, "Survey of sampling-based methods for uncertainty and sensitivity analysis," *Reliability Engineering & System Safety*, vol. 91, no. 10-1, pp. 1175 – 1209, 2006, the Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004) {SAMO}. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0951832005002292>
- [8] W. S. Humphrey, *A Discipline for Software Engineering*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
- [9] A. Finkelstein, M. Harman, S. Mansouri, J. Ren, and Y. Zhang, "'Fairness analysis' in requirements assignments," in *International Requirements Engineering, 2008. RE'08. 16th IEEE*, Sept 2008, pp. 115–124.
- [10] 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 (GECCO'07)*. New York, NY, USA: ACM, 2007, pp. 1129–1137. [Online]. Available: <http://doi.acm.org/10.1145/1276958.1277179>
- [11] A. Budzier, "Why your it project may be riskier than you think," *Harvard Business Review*, vol. 89, no. 9, pp. 23–25, 2011.
- [12] H.-G. Beyer and B. Sendhoff, "Robust optimization—a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33, pp. 3190–3218, 2007.
- [13] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, Aug. 2011. [Online]. Available: <http://dx.doi.org/10.1137/080734510>
- [14] M. Harman, J. Krinke, J. Ren, and S. Yoo, "Search based data sensitivity analysis applied to requirement engineering," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*. New York, NY, USA: ACM, 2009, pp. 1681–1688. [Online]. Available: <http://doi.acm.org/10.1145/1569901.1570126>
- [15] M. Paixão and J. Souza, "A scenario-based robust model for the next release problem," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO'13)*. New York, NY, USA: ACM, 2013, pp. 1469–1476. [Online]. Available: <http://doi.acm.org/10.1145/2463372.2463547>
- [16] P. Festa, "A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems," in *Transparent Optical Networks (ICTON), 2014 16th International Conference on*, July 2014, pp. 1–20.
- [17] Y. Zhang, *Multi-Objective Search-based Requirements Selection and Optimisation*, 2010.
- [18] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *IEEE Softw.*, vol. 14, no. 5, pp. 67–74, Sep. 1997. [Online]. Available: <http://dx.doi.org/10.1109/52.605933>
- [19] S. L. Lim, D. Damian, and A. Finkelstein, "Stakesource2.0: Using social networks of stakeholders to identify and prioritise requirements," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: ACM, 2011, pp. 1022–1024. [Online]. Available: <http://doi.acm.org/10.1145/1985793.1985983>
- [20] Y. Zhang, A. Finkelstein, and M. Harman, "Search based requirements optimisation: Existing work and challenges," in *Requirements Engineering: Foundation for Software Quality*, ser. Lecture Notes in Computer Science, B. Paech and C. Rolland, Eds. Springer Berlin Heidelberg, 2008, vol. 5025, pp. 88–94.
- [21] D. Greer and G. Ruhe, "Software release planning: an evolutionary and iterative approach," *Information and Software Technology*, vol. 46, no. 4, pp. 243–253, 2004.
- [22] M. Harman, A. Skaliotis, K. Steinhöfel, and P. Baker, "Search-based approaches to the component selection and prioritization problem," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '06. New York, NY, USA: ACM, 2006, pp. 1951–1952. [Online]. Available: <http://doi.acm.org/10.1145/1143997.1144321>
- [23] H. Jiang, J. Zhang, J. Xuan, Z. Ren, and Y. Hu, "A hybrid aco algorithm for the next release problem," in *Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on*, June 2010, pp. 166–171.
- [24] Y. Zhang, M. Harman, G. Ochoa, G. Ruhe, and S. Brinkkemper, "An empirical study of meta-and hyper-heuristic search for multi-objective release planning," *RN*, vol. 14, p. 07, 2014.
- [25] N. Veerapen, G. Ochoa, M. Harman, and E. K. Burke, "An integer linear programming approach to the single and bi-objective next release problem," *Information and Software Technology*, vol. 65, pp. 1 – 13, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950584915000658>
- [26] M. Harman, J. Krinke, I. Medina-Bulo, F. Palomo-Lozano, J. Ren, and S. Yoo, "Exact scalable sensitivity analysis for the next release problem," *ACM Trans. Softw. Eng. Methodol.*, vol. 23, no. 2, pp. 19:1–19:31, Apr. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2537853>
- [27] E. Letier, D. Stefan, and E. T. Barr, "Uncertainty, risk, and information value in software requirements and architecture," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 883–894. [Online]. Available: <http://doi.acm.org/10.1145/2568225.2568239>
- [28] Y. Li, J. Chen, and L. Feng, "Dealing with uncertainty: A survey of theories and practices," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2463–2482, Nov 2013.
- [29] G. Bammer and M. Smithson, *Uncertainty and risk: multidisciplinary perspectives*. Routledge, 2012.
- [30] S. Rasoulilian and L. A. Ricardez-Sandoval, "Uncertainty analysis and robust optimization of multiscale process systems with application to epitaxial thin film growth," *Chemical Engineering Science*, vol. 116, pp. 590 – 600, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0009250914002371>
- [31] D. Chaffart, S. Rasoulilian, and L. A. Ricardez-Sandoval, "Distributional uncertainty analysis and robust optimization in spatially heterogeneous multiscale process systems," *AIChE Journal*, pp. n/a–n/a, 2016. [Online]. Available: <http://dx.doi.org/10.1002/aic.15215>
- [32] M. D. McKay, "Sensitivity and uncertainty analysis using a statistical sample of input values," *Uncertainty analysis*, pp. 145–186, 1988.
- [33] G. Bayraksan, "Monte carlo sampling-based methods in stochastic programming," Ph.D. dissertation, University of Texas at Austin, 2005.
- [34] A. Saltelli, K. Chan, and E. Scott, *Sensitivity Analysis*, ser. Wiley paperback series. Wiley, 2009, no. no. 2008. [Online]. Available: <http://books.google.co.uk/books?id=gOcePwAACAAJ>
- [35] E. Smith, "Uncertainty analysis," *Encyclopedia of environmetrics*, 2006.
- [36] Z. Zi, "Sensitivity analysis approaches applied to systems biology models," *Systems Biology, IET*, vol. 5, no. 6, pp. 336–346, Nov 2011.
- [37] S. Wagner, "Global sensitivity analysis of predictor models in software engineering," in *Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop on*. IEEE, 2007, pp. 3–3.
- [38] A. L. Soyster, "Technical note: convex programming with set-inclusive constraints and applications to inexact linear programming," *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.21.5.1154>
- [39] M. Li, S. Azarm, and V. Aute, "A multi-objective genetic algorithm for robust design optimization," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO'05)*. New York, NY, USA: ACM, 2005, pp. 771–778. [Online]. Available: <http://doi.acm.org/10.1145/1068009.1068140>
- [40] M. Papadrakakis, N. Lagaros, and V. Plevris, "Design optimization of steel structures considering uncertainties," *Engineering Structures*, vol. 27, no. 9, pp. 1408 – 1418, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0141029605001483>
- [41] W. Heaven and E. Letier, "Simulating and optimising design decisions in quantitative goal models," in *Requirements Engineering Conference (RE), 2011 19th IEEE International*. Aug 2011, pp. 79–88.
- [42] I. Das, "Robustness optimization for constrained nonlinear programming problems," *Engineering Optimization*, vol. 32, no. 5, pp. 585–618, 2000. [Online]. Available: <http://dx.doi.org/10.1080/03052150008941314>
- [43] L. Li, M. Harman, E. Letier, and Y. Zhang, "Robust next release problem: Handling uncertainty during optimization," in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, ser. GECCO '14. New York, NY, USA: ACM, 2014, pp. 1247–1254.