Robust Next Release Problem: Handling Uncertainty During Optimization

Lingbo Li
University College London
Malet Place, London
WC1E 6BT, UK
lingbo.li.13@ucl.ac.uk

Mark Harman
University College London
Malet Place, London
WC1E 6BT, UK
mark.harman@ucl.ac.uk

Emmanuel Letier
University College London
Malet Place, London
WC1E 6BT, UK
e.letier@ucl.ac.uk

Yuanyuan Zhang
University College London
Malet Place, London
WC1E 6BT, UK
yuanyuan.zhang@ucl.ac.uk

ABSTRACT

Uncertainty is inevitable in real world requirement engineering. It has a significant impact on the feasibility of proposed solutions and thus brings risks to the software release plan. This paper proposes a multi-objective optimization technique, augmented with Monte-Carlo Simulation, that optimizes requirement choices for the three objectives of cost, revenue, and uncertainty. The paper reports the results of an empirical study over four data sets derived from a single real world data set. The results show that the robust optimal solutions obtained by our approach are conservative compared to their corresponding optimal solutions produced by traditional Multi-Objective Next Release Problem. We obtain a robustness improvement of at least 18% at a small cost (a maximum 0.0285 shift in the 2D Pareto-front in the unit space). Surprisingly we found that, though a requirement's cost is correlated with inclusion on the Pareto-front, a requirement's expected revenue is not.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements—*Specifications Methodologies*

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Software engineering, Genetic algorithms, Multi-objective optimization, Robustness of solutions, Empirical study

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598334.

1. INTRODUCTION

Uncertainty is an inherent characteristic of software engineering, and cannot be avoided [25]. The uncertainties in software engineering include uncertainty about the actual implementation cost of a software project, actual revenue of implemented features, and the possibility that the resulting product achieves its expected economic performance. Underestimated or ignored uncertainties may bring risks into software projects.

Previous Multi-Objective Next Release Problem (MONRP) work was concerned with point-based estimation [11,12,23]. The limitation of point-based estimation is that uncertainty and risk are either underestimated or completely overlooked [8]. For example, a requirement engineer may estimate that the *expected cost* of a feature is £500 and *expected revenue* is £700. The *expected net-revenue* (*expected revenue − expected cost*) is £200. There is a risk that the *expected net-revenue* may be lower than a threshold assigned by decision makers due to uncertainty concerning the true revenue and cost. The development cost of the feature may exceed £500, and the revenue of the feature may lower than £700. This tendency for feature attributes to change has been regarded as one of reasons why software development is difficult and expensive [22].

In order to mitigate the impact of uncertainty, previous work on requirements engineering undertook sensitivity analysis after optimizing the Next Release Problem (NRP) [14, 16]. Sensitivity analysis was performed on solutions to investigate solutions's uncertainty. The most related work to our study is Paixão and Souza's work [20]. They formulated the uncertainty of NRP into a scenario-based framework. In their approach, the revenue of requirements have the same probability distribution: the cost of each requirement is quantified in a deterministic continuous interval format. Though this approach addresses uncertainty, it is limited by the implicit assumption that the real probability is universal.

In this paper, we adopt a search-based optimization technique with Monte-Carlo Simulation (MCS) to address uncertainty and risk in the early stages of the software engineering development process. Our approach makes explicit the trade-off between uncertainty/risk and traditional attributes of cost & revenue. It is assumed that the Probability Density Function (PDF) of features (in terms of cost and revenue) has been determined by a prior risk analysis [7]. The paper builds novel formulations of uncertainty to guide the NRP and presents robust Pareto-optimal solutions to decision makers. There are two notions of uncertainty measurement introduced: size of uncertainty region [18] and the failure possibility: the probability that actual cost exceeds a threshold. There are two definitions of robust solution considered in the paper: 1) the solution's *payoff* (in terms of cost and revenue) has narrow fluctuation range, 2) the actual cost of solution has low possibility to exceed the threshold. Each is 'robust' in the sense that it minimizes the risk associated with a requirement choice. We compute the uncertainties of variables as probability distributions, and simulate them by MCS. We measure the two kinds of robustness, and explore the Pareto-front by using multi-objective evolution algorithm. Our approach can provide the solutions that balance the trade-off among revenue, cost, and robustness in a software project.

The main contributions of this paper are the following:

1. The paper introduces two notions of uncertainty measurements for NRP: *MCNRP-US* (MCS for NRP - Uncertainty Size) and *MCNRP-R* (MCS for NRP - Risk) (Section 3). Each formulation has three objectives for optimization.

2. The paper is the first paper that addresses robust NRP by integrating the MCS into the search process itself. Previous work treats it as a post hoc computation. As a result, the decision maker has no option to *minimize* uncertainty, which is supported by our approach.

3. Results of our evaluation on 4 data sets derived from the Motorola data set [4] indicate that our approach reduces risk/uncertainty with very little change to the traditional 2D MONRP Pareto-front.

The structure of rest of the paper is organized as follows: Section 2 describes the related work in NRP, robust optimization and sensitivity analysis. Section 3 formally defines the research problem. Section 4 describes the Monte-Carlo Simulation & search algorithm used. Section 5 describes the experimental setup, algorithm configuration, evaluation methods, and research questions. Section 6 reports the results of the experiments and analyses the findings. Section 8 presents the conclusion and suggestions for future work.

2. RELATED WORK

The term NRP was introduced by Bagnall et al. [3] in 2001. The NRP aims to search the feasible and ideal solution set to balance the requests from different stakeholders. The NRP was formulated as a single-objective optimization problem by Bagnall et al. [3]. Zhang et al. [23] extended the NRP to MONRP by formulating revenue and cost as two objectives.

In terms of uncertainty analysis, researchers usually adapt quantitative analysis methods, such as uncertainty analysis [24], to evaluate the robustness of the model. Sensitivity analysis is one uncertainty analysis method. Harman et al. [16] studied the data sensitivity of NRP and MONRP by using a local sensitivity analysis approach "One-At-a-Time" [21]. This approach measured data sensitivity by perturbing variables upward or downwards to try out various what-if scenarios. To avoid potential noise from the inherent character of stochastic of the meta-heuristic algorithm, Harman et al. [14] applied an exact algorithm, a variant of the Nemhauser-Ullmann's algorithm [19], to study the precise sensitivity analysis of NRP. Al-Emran and Ruhe et al. [1, 2] applied probabilistic sensitivity analysis which integrates MCS with process simulation to study the impact of uncertainty in operational release planning and product release planning as a post-analysis.

However, Hans-Georg and Bernhard [6] indicated that it is important to investigate uncertainty *during* the process of optimization rather than using post-analysis. They proposed robust optimization [6]. Li [18] introduced a novel metric of uncertainty for guiding multi-objective optimization problem. In his work, the uncertainty of a parameter's true value was represented as an interval and the uncertainty of solutions was represented as tolerance region (uncertainty size). In 2013, Paixão and Souza used a scenario-based robust optimization framework for NRP to produce robust optimal solutions [20].

Our paper is the first paper on Search-Based Software Engineering (SBSE) [15] to introduce MCS to simulate the uncertainties of NRP as one of the objectives to guide the search to explore the robust Pareto-optimal front.

3. PROBLEM FORMULATION

In this section, we describe the definition of the NRP and the metrics that capture uncertainty in our approach.

3.1 NRP Problem

It is assumed 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 Eq.1 and the set of possible requirements is denoted by Eq.2.

$$C = \{c_1, \cdots, c_m\} \quad (1)$$

$$R = \{r_1, \cdots, r_n\} \quad (2)$$

where m is the number of stakeholders, and n is the number of features.

In this paper, all requirements are independent of each other. 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 fulfill the requirement as given by Eq.3.

$$Cost = \{cost_1, \cdots, cost_n\} \quad (3)$$

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 Eq.4:

$$Weight = \{w_1, \cdots, w_m\} \quad (4)$$

Subject to: $w_j \in [0, 1]$, and $\sum_{j=1}^{m} w_j = 1$.

It is assumed that the importance of each requirement for each stakeholder is 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 assump-

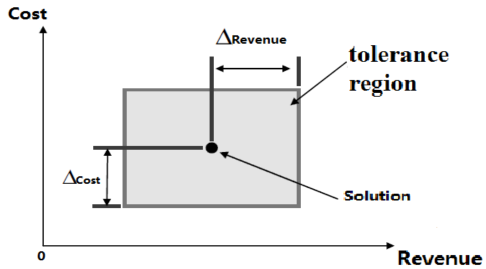


Figure 1: The tolerance region of a MONRP solution [18].

tion, 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 Eq.5.

$$Revenue_i = \sum_{j=1}^{m}(w_j \cdot value(r_i, c_j)) \quad (5)$$

In NRP, the solution is presented as a decision vector $\vec{x} = \{x_1, \cdots, x_n\} \in \{0, 1\}$ 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.

3.2 Robust MONRP formulation

This paper considers two types of robustness in MONRP. These two definitions of robust solution are "reduction of the uncertainty size", and "reduction of the possibility that actual cost exceeds a threshold" (named "failure risk reduction").

3.2.1 Uncertainty Size Reduction (MCNRP-US)

Uncertainty size is used to measure the tolerance region of the solutions of multi-objective optimization problem in d dimensions (d is the number of the objectives). For example, in NRP, $\Delta cost_i$ is an acceptable fluctuation range of the cost of the ith requirement. The tolerance region consists of the confidence levels of each fitness value. The confidence level indicates the most likely fluctuation range of fitness values. Figure 1 illustrates a tolerance region for a NRP solution with cost and revenue objective functions. The shaded area is the tolerance region of the given solution. In Li's work [18], the standard deviation of each fitness value is used as confidence level. Hence, the tolerance region is composed of the standard deviation of each fitness value.

The size of tolerance region is presented by its normalized hyper-perimeter (Eq.6) and hyper-volume (Eq.7). To normalize the metric of each fitness value, we need to define fitness value referent with respect to each objective function.

$$perimeter(\vec{x}) = \sum_{k=1}^{d} \frac{2 \cdot \Delta fitness_k(\vec{x})}{referent\_fitness_k} \quad (6)$$

$$volume(\vec{x}) = \prod_{k=1}^{d} \frac{2 \cdot \Delta fitness_k(\vec{x})}{referent\_fitness_k} \quad (7)$$

where d is the number of objective functions. Therefore, all our fitness values lie in a normalized unit space. This facilitates comparison of Pareto-front using Euclidean Distance.

Besides, the weighted sum of these two metrics is defined as the *uncertainty size* and shown in Eq.8

$$Size(\vec{x}) = \alpha \cdot volume(\vec{x}) + \beta \cdot perimeter(\vec{x}) \quad (8)$$

Where $\alpha + \beta = 1$. In this work, we defined $\alpha = 0.5$ and $\beta = 0.5$.

We named this model as *MCNRP-US* (MCS for NRP-Uncertainty Size). The *MCNRP-US* consisting of the objective functions can be presented as follows (Eq.9, Eq.10, and Eq.11):

$$Maximize f_1(\vec{x}) = \sum_{i=1}^{n}(x_i \cdot Expected\_Revenue_i) \quad (9)$$

$$Minimize f_2(\vec{x}) = \sum_{i=1}^{n}(x_i \cdot Expected\_Cost_i) \quad (10)$$

$$Minimize f_3(\vec{x}) = Size(\vec{x}) \quad (11)$$

3.2.2 Failure Risk Reduction (MCNRP-R)

In our approach, the risk of a given solution is measured by the probability that the actual cost exceeds a threshold determined by the decision maker. In order to reduce the risk of budget overrun, our second approach minimizes the probability that actual cost exceeds the budget (Eq.12).

$$Risk(\vec{x}) = Pro(actual\_cost(\vec{x}) > \theta \cdot Expected\_Cost(\vec{x})) \quad (12)$$

Where θ is the percentage assigned by the decision maker (e.g., $\theta = 150\%$), and Pro means Probability.

This model named *MCNRP-R* (MCS for NRP-Risk). The objective functions of *MCNPR-R* are shown as Eq.9, Eq.10, and Eq.13:

$$Minimize f_3(\vec{x}) = Risk(\vec{x}) \quad (13)$$

4. OPTIMIZATION APPROACH

Our approach contains two procedures: MCS and multi-objective optimization. MCS enables us to simulate and evaluate a large number of scenarios effectively. The output of MCS process is used by the multi-objective optimization process. The multi-objective optimization is used to optimize multiple and possibly conflicting objectives simultaneously. In this paper, we adopt the NSGA-II algorithm for optimization.

Monte Carlo Simulation (MCS) [13] is a computerized mathematical technique to explore the range of possible outcomes of the model and the probability that these outcomes will occur. The principle of MCS is to sample a large number of scenarios generated by substituting the probability distributions of model parameter values. It then calculates the results of model for all scenarios.

MCS generates a "scenarios database": an $s \times n$ matrix, *Simulations*, where s is the number of scenarios and n is the number of requirements. The element $Simulations[i, j]$ denotes the value of requirement j in ith scenario. The value includes the simulated revenue and the simulated cost of a given requirement. The number of scenarios was set to $10,000$.

The well-known Non-dominated Sorting Genetic Algorithm-II (NSGA-II) was introduced by Deb et al. [9]. We use

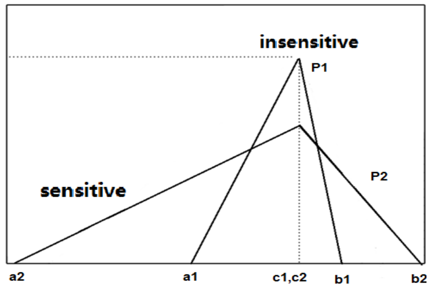


Figure 2: Illustration of the triangle probability distribution and the classification of sensitive and insensitive distributions [10]. c_1 and c_2 are the mode value of probability distribution P_1 and probability distribution P_2, respectively. a_1 and b_1 is the lowest value and highest of P_1 respectively while a_2 and b_2 is the lowest value and highest of P_2. P_1 is considered to be more 'stable' (insensitive).

NSGA-II to provide a Pareto-front that captures the trade-off between cost & revenue and risk (assessed using MCS).

5. EXPERIMENTAL SET UP

5.1 Data Sets

There are four synthetic data sets used in our experiments. The four data sets are synthetically constructed from one real project data set from Motorola [4]. The Motorola data set concerns a set of 35 requirements for hand held communication devices. Each requirement has the estimated implementation cost and expected revenue level. There is no uncertainty information for the cost and revenue of requirements. Our approaches can accept most kinds of probability distributions, such as uniform distribution, normal distribution, and discrete distribution. In this work, we simulated these uncertainties according to the "triangle probability distribution" illustrated in Figure 2.

Our four synthetic data sets represent four general scenarios (*S1* - *S4*), according to the degree of uncertainty about requirements' cost.

S1 Requirements for low cost have low probability to change (insensitive), while requirements for high cost have high probability to change (sensitive).

S2 Requirements for low cost have high probability to change (sensitive), while requirements for high cost have low probability to change (insensitive).

S3 Requirements for low cost have low probability to change (insensitive), while requirements for high cost have low probability to change (insensitive).

S4 Requirements for low cost have high probability to change (sensitive), while requirements for high cost have high probability to change (sensitive).

The ith requirement would be classified as low cost requirement if $cost_i < \frac{\sum_{j=1}^{n} cost_j}{n}$, otherwise high cost requirement, where n is the number of requirements. We define low probability as the possible change range is within 100%, while for high probability it is within 250%. The uncertainty of each cost is stochastically generated based on the above definitions. The uncertainty of revenue is randomly generated to have low probability (insensitive). A partial data of *S1* reported in Table 1.

Table 1: Illustrative fragment of *S1* data

	Cost			Revenue			
NAME	Mode	Min	Max	Mode	Min	Max	Sensitivity
REQ 1	100.00	79.42	127.91	3.00	0.65	3.32	insensitive
REQ 2	50.00	15.08	53.51	3.00	1.30	3.95	insensitive
REQ 3	300.00	270.74	1154.15	3.00	0.32	4.76	sensitive
REQ 4	80.00	52.73	105.30	3.00	1.31	5.50	insensitive
REQ 5	70.00	42.00	78.77	3.00	1.66	4.62	insensitive
REQ 6	100.00	87.34	133.04	3.00	1.01	4.19	insensitive
REQ 7	1000.00	620.75	3671.35	3.00	0.77	5.68	sensitive

5.2 Search Algorithmic Tuning

We base our algorithmic parameter & tuning on those used in previous work on MONRP [23]. We used binary encoding to represent the decision vector. The initial population size was set to 500. The algorithm was run for a maximum of $50,000$ function evaluations. The genetic operators used in our approaches are tournament selection (with tournament size of 5), single-point crossover (with crossover probability 0.8) and bitwise mutation (with the mutation probability $1/n$ where n is the number of requirements). The algorithm was executed 30 times for each data set, to cater for the stochastic nature of the algorithm.

5.3 Evaluation

5.3.1 *Price of Robustness*

In order to measure such loss between the proposed robust Pareto-front and original Pareto-front with regard to cost and revenue objectives, we utilized the "reduction factor" [5] to measure the "Price of Robustness". This factor measures the distance between two fronts [16]. To compute such distance, we defined $(A_1, \cdots, A_p)$ as the $front_a$ which contains p solutions, while $(B_1, \cdots, B_q)$ denotes the q solutions in $front_b$, where p and q are the number of solutions in $front_a$ and $front_b$ respectively.

The distance from solution A to solution B is computed by the normalized objective values and Euclidean Distance. In the case of "Price of Robustness", the distance between solution A and B is defined as:

$$Dis(A, B) = \pm\sqrt{\sum_{i=1}^{d}(A\_fit_i - B\_fit_i)^2} \qquad (14)$$

Where d is the number of objectives. A_fit_i and B_fit_i are the ith objectives value of A and B, respectively.

The distance from solution A to geometrically closest solution B on $front_b$ is presented as the distance from solution A to $front_b$ (Eq.15).

$$Dis(A, front_b) \quad = \quad Dis(A, B) \qquad (15)$$

Therefore, the distance from $front_a$ to $front_b$ is the mean value of the distance from every solution on $front_a$ to $front_b$.

$$Dis(front_a, front_b) \quad = \quad \frac{\sum_{i=1}^{p} Dis(A_i, front_b)}{p} \qquad (16)$$

5.3.2 Probabilistic Sensitivity Analysis

To measure the amount of robustness improvement achieved by our robust optimization approach, we performed a probabilistic sensitivity analysis. Firstly, we used the same sampling technique to simulate the uncertainties of data. After that, we adopted robustness formulations defined in our paper to calculate the robustness of Pareto-optimal solutions of traditional approach.

5.4 Research Questions

In order to evaluate the effectiveness and usefulness of the approaches, we carried out two experimental studies to assess the efficiency of the approaches and four scenarios to evaluate its usefulness. In the experiments, we compared the results obtained from our approaches with the ones obtained from MONRP, and formalized one research questions. The question is whether the proposed approaches can provide more robust solutions to decision makers with less sacrifice? This question formulated into three more detailed sub-questions (***RQ1***, ***RQ2***, and ***RQ3***).

Additionally, to aid the decision making support before preforming such professional tools, this paper also investigated the correlations between attributes of a requirement and its inclusion in solutions on the Pareto-front. This is formulated into the fourth question ***RQ4***.

RQ1 Do the proposed two kinds of robust optimization improve robustness? This question will be answered by analyzing and comparing the robustness of solutions which were generated by our approaches and the original MONRP.

RQ2 How much "Price of Robustness" would be paid for the proposed robust optimization approaches? We will answer this question by calculating the distance between the Pareto-front obtained from our approaches and those obtained from the original MONRP. The distance was used to measure the loss in *payoff*.

RQ3 How similar are the Pareto-fronts produced by our new approaches and the one produced by traditional MONRP? We computed and ranked the proportion of requirements being selected in solutions on the Pareto-front. Then we used Kendall's τ correlation coefficient to statistically investigate the degree of similarity between the rankings of requirements included in solutions on the Pareto-front to answer this question.

RQ4 Which attributes of a requirement are correlated with inclusion in solutions on the Pareto-front?

We performed an intuitive analysis to answer the ***RQ1*** and ***RQ2***, while more statistically analysis to answer the ***RQ3*** and ***RQ4***.

6. EXPERIMENTAL RESULTS & ANALYSIS

This section presents two different robust models and the results of applying these two models on four synthetic problem instances. Two experiments were conducted and the illustrations of results are presented in Figures 3 and 4 (Figures 3a, 3b, 3c, 3d, and Figures 4a, 4b, 4c, 4d), for *E1* & *E2* respectively. The two experiments, *E1* & *E2*, are described below:

Table 2: The Robustness & Comparison of the *MCNRP-US* Approach and the Traditional Approach

	S1	*S2*	*S3*	*S4*
MCNRP-US	0.1531	0.1558	0.1850	0.1290
Original Approach	0.1983	0.1599	0.1993	0.1511
Price of Robustness	0.0110	0.0201	0.0154	0.0102
Robustness Improvement	22.78%	2.54%	7.19%	14.65%

E1 The first experiment aims at evaluating the *MCNRP-US* approach and the "Price of Robustness" of this approach, when the decision maker expects to obtain robust solutions within a defined fluctuation range.

E2 The second experiment evaluates the *MCNRP-R* approach and its "Price of Robustness" for the situation in which the decision maker would like to acquire robust solutions which have a low risk of budget overrun.

In order to compare our proposed approach to the traditional MONRP approach, the Pareto-fronts of proposed approach are presented by dark black patterns and traditional ones by grey (red when viewed in colour) patterns. This selection quantitatively analysis and answer the *RQ3* and *RQ4* as well.

6.1 Experiment One (E1)

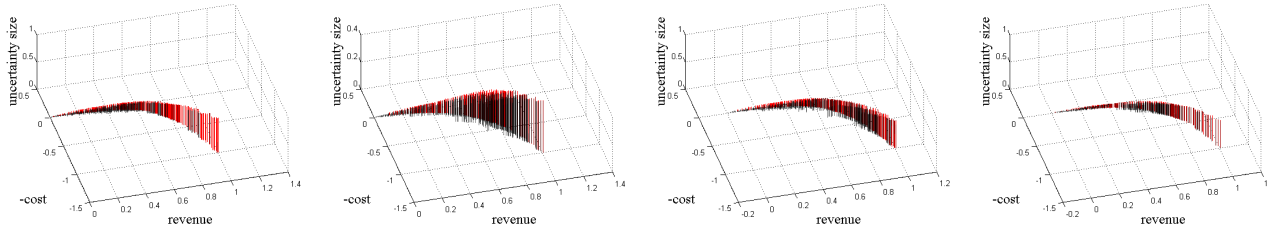
In *E1*, the *uncertainty size* of a solution is taken into account. The results of *E1* are shown in Figures 3a, 3b, 3c, and 3d corresponding to scenarios *S1*, *S2*, *S3*, and *S4*, respectively. The figures illustrate the three-dimensional Pareto surface. Each bar represents a solution on the Pareto-front. The location of each bar in the *cost-revenue* plane presents the *cost* and *revenue* of the solution respectively. The height of the bar shows the *uncertainty size* for each solution.

From the results of *E1* for *S1*, *S2*, *S3*, and *S4*, we observe that, as the overall fulfilled *cost* increases, the *uncertainty size* of solution also increases. We also observe that there are minor differences between the Pareto-fronts of *MCNRP-US* and the traditional approach in *S1* and *S4* (Figures 3a and 3d), while there are larger differences between *S2* and *S3* (Figures 3b and 3c). High cost requirements naturally have more impact on solution sensitivity than low cost requirements [16]. Requirements with high cost are stable in *S2* and *S3*, and the proposed first approach tends to select the "stable" solution rather than the solutions just have good economic performance but "unstable".

Table 2 presents the results of probabilistic sensitivity analysis for *E1*, the "*Price of Robustness*" for "*MCNRP-US*" approach, and how much robustness with regarded to *uncertainty size* improved by applying this approach.

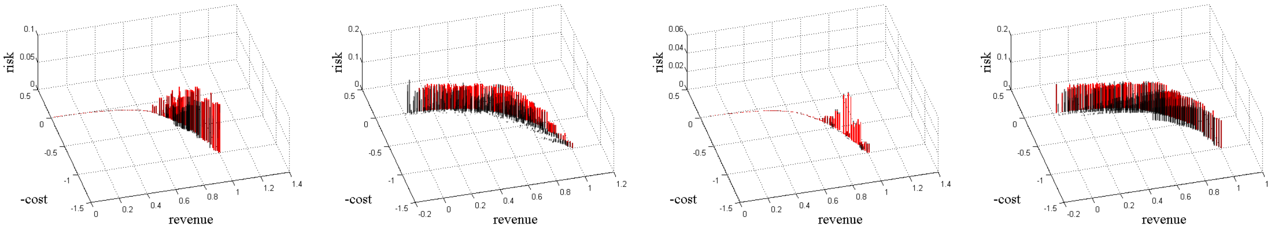
Based on the results in this table we answer ***RQ1*** and ***RQ2*** (for *MCNRP-US*) as follows: On average, the *MCNRP-US* generates more robust solutions with respect to *uncertainty size*. The overall improvement is not large: after normalizing *cost* and *revenue*, the magnitude of standard deviation of *cost* and *revenue* is small, so the magnitude of *uncertainty size* is small. Even so, it is interesting that the robustness improvements for *S1* and *S4* (22.78% and 14.65% respectively) are better than the improvements for *S2* and *S3* (2.54% and 7.19% respectively).

Although the improvement of *MCNRP-US* is not dramatic, it pays a little as the "*Price of Robustness*". Therefore, we conclude that applying our *MCNRP-US* approach,



(a) The results of *E1* in *S1* (b) The results of *E1* in *S2* (c) The results of *E1* in *S3* (d) The results of *E1* in *S4*

Figure 3: The Pareto-front of *MCNRP-US* and Original Approach



(a) The results of *E2* in *S1* (b) The results of *E2* in *S2* (c) The results of *E2* in *S3* (d) The results of *E2* in *S4*

Figure 4: The Pareto-front of *MCNRP-R* and Original Approach

Table 3: The Robustness & Comparison of the *MCNRP-R* Approach and the Traditional Approach

	S1	*S2*	*S3*	*S4*
MCNRP-R	0.0396	0.0404	0.0109	0.0591
Original Approach	0.0500	0.0755	0.0132	0.0888
Price of Robustness	0.0036	0.0253	0.0003	0.0285
Robustness Improvement	20.82%	46.49%	17.70%	33.37%

the decision maker pays a small price to obtain a more robust Pareto-front, whose solutions have smaller *uncertainty size*.

6.2 Experiment Two (E2)

The results of *E2* are plotted in Figures 4a, 4b, 4c, and 4d. In *E2*, the *risk* was considered as a third objective.

From the results of *E2* in *S1*, *S2*, *S3*, and *S4*, a general trend is observed: the degree of *risk* increases as overall *cost* increases. However, there is an interesting observation in Figure 4b. The *risk* is inversely proportional to *cost*. The reason for this phenomenon is that the *risk* is directly proportional to the stability of the probability distribution of *cost*. The more stable the probability distribution is, the lower risk there will be. In *E2*, there are some other interesting observations: According to the results, we observe that the obtained "robust" Pareto-fronts are quite close to those obtained from original MONRP in *S1* and *S3*, while there are a big gap in *S2* and *S4*. This is because the probability distribution of high *cost* is unstable in *S2* and *S4*.

Table 3 shows that the robustness with regards to risk can be noticeably improved by the *MCNRP-R* approach compared to traditional approach. Moreover, the payment (*Price of Robustness*) is low.

As an overall answer ***RQ1*** and ***RQ2*** (for MONRP-R) we find that we can achieve an improvement of at least 18% in robustness with only a little change in 2D cost-revenue Pareto-front (maximum 0.0285 in a unit space). That is, the penalization due to robustness is very small for all scenarios, which qualifies the effectiveness of *MCNRP-R* approach.

Table 4: The Correlation of Rankings of Requirements

		MONRP&R	MONRP&US	US&R
S1	τ	0.9361	0.7345	0.7311
	p-value	< 0.000	< 0.000	< 0.000
S2	τ	0.8646	0.7872	0.8756
	p-value	< 0.000	< 0.000	< 0.000
S3	τ	0.9655	0.7233	0.7311
	p-value	< 0.000	< 0.000	< 0.000
S4	τ	0.8646	0.8713	0.8387
	p-value	< 0.000	< 0.000	< 0.000

In this table, R means MONRP-R, and US means MONRP-US.

6.3 Statistical Analysis

To answer the ***RQ3*** and ***RQ4*** statistically, Kendall's τ correlation coefficient τ is used to quantitatively analyse the correlation between and within the approaches. Table 4 shows Kendall's τ correlation coefficient and corresponding p-value calculated for the relation between the paired approach (MONRP and MONRP-R, MONRP and MONRP-US, and MONRP-R and MONRP-US) with regard to each scenario. If all solutions in Pareto-front agree on a requirement to be selected, the requirement is said to be "closed" [17]. Here, we generalize this notion of "closed" decision to investigate correlations between degrees of "closedness".

The Table 4 reveals that there are existing strong correlations between the rankings of requirements produced by each approach on each scenario. All τ coefficients are greater than 0.7, and p-values are very close to zero. This confirms that

Table 5: The Correlation between the Attributes of Requirement and its Ranking

	Cost		Revenue		R/C	
MONRP	τ	$p-$Value	τ	$p-$Value	τ	$p-$Value
S1	-0.7748	< 0.000	0.0723	0.55358	0.9597	< 0.000
S2	-0.7569	< 0.000	0.1413	0.23846	0.9521	< 0.000
S3	-0.7771	< 0.000	0.074	0.54138	0.9521	< 0.000
S4	-0.7704	< 0.000	0.1346	0.26185	0.9554	< 0.000
MONRP-US	τ	$p-$Value	τ	$p-$Value	τ	$p-$Value
S1	-0.5899	< 0.000	0.0824	0.49827	0.721	< 0.000
S2	-0.6034	< 0.000	0.2336	0.049495	0.7714	< 0.000
S3	-0.5832	< 0.000	0.0924	0.44599	0.7008	< 0.000
S4	-0.6807	< 0.000	0.1765	0.14052	0.8521	< 0.000
MONRP-R	τ	$p-$Value	τ	$p-$Value	τ	$p-$Value
S1	-0.7244	< 0.000	0.1092	0.3661	0.8958	< 0.000
S2	-0.6807	< 0.000	0.1966	0.09972	0.8555	< 0.000
S3	-0.758	< 0.000	0.0924	0.44599	0.9294	< 0.000
S4	-0.674	< 0.000	0.1899	0.11213	0.8521	< 0.000

In this table, Cost is the Expected Cost, Revenue is the Expected Revenue, and R/C is the Expected *Revenue-to-Cost* Ratio.

the rankings of requirements produced by each approach are similar to each other.

Hence, the Pareto-fronts on Cost-Revenue dimension generated by each approach are similar to each other. We further observe that the correlation is stronger between MONRP and MONRP-R than MONRP and MONRP-US. This answers ***RQ3***.

In order to answer ***RQ4***, Table 5 uses Kendall's tau correlation analysis to statistically describe the correlation between the attributes of requirements and the rankings of requirements. The results reveal that, in general, the requirement's *Revenue-to-Cost* ratio and *Cost* have strong monotonic correlation with its likelihood of inclusion, while its *Revenue* is uncorrelated. The requirement's *Revenue-to-Cost* ratio is the most strongly correlated.

7. THREATS TO VALIDITY

In this paper, our approach applies a multi-objective search-based technique, which is augmented with MCS, to solve the NRP in the presence of uncertainty. We evaluated our work respecting to its construct validity, internal validity, and external validity.

7.1 Construct validity

The threat to construct validity is concerning whether our approach can deal with the real uncertainty. Our method utilizes MCS to compute the estimation uncertainties on requirements, and to produce thousands of scenarios to simulate the actual uncertain NRP parameters. In this paper, only triangle probability distribution is used to present the uncertainties, while there are sorely other kinds of uncertainties that might be used in risk analysis. Moreover, the kinds of uncertainties are used to generate the scenarios whose attributes are formed as input parameters of MONRP approach. Hence, it is straightforward to introduce other kinds of uncertainties to model the uncertainties of requirements. Our approach aims to handle the uncertainty of NRP by MCS. MCS can simulate most kinds of uncertainties. Besides, the types of uncertainties can be formed into the requirements by engineers. Therefore, we believe our approach can be extended to solve the real NRP with various uncertainties.

7.2 Internal validity

The internal validity is concerned with any possible factor that may perturb the experimental evaluations. Hence, in our experiments, we exclude other system applications, so the experimental machine only runs our application. Each experiment was repeated 30 times to cater for variations in algorithmic performance. The other potential threat to internal validity concerns the algorithm parameters tuning that could have affected the experimental results during the evaluations. Our work is based on the works of Zhang et al. [23], and the algorithm parameters have been previously studied [11] [22].

7.3 External validity

In the experimental study, we evaluated our approaches over four data sets derived from one single real world data set from Motorola. There is no uncertainty information within the Motorola data set. Therefore we simulated the four scenarios with uncertainties based on different degrees. Since we lack a real software engineering problem data set, the scale of our experiments and number of scenarios remains insufficient. More work is required to analyse different scenarios & models of uncertainty.

In our work, the experimental results did not show any critical threat to its scalability. Our implementation is able to process about $10,000$ scenarios in 15s, and each scenario consists of $50,000$ runs. The heaviest computational consumption is MCS. While the scale of problem is increasing, our methods become more complex and time consuming. A simple way to address this issue is to reduce the number of simulations. Consequently, the computation time will reduce, while the simulation error will increase. Since this paper focusses on proposing a novel approach to handle uncertainty in NRP, optimizing the performance of this approach will be an interesting further work.

8. CONCLUSIONS & FUTURE WORK

In this paper, we introduced an MCS based robust optimization approach for requirement analysis and optimization.

We introduced two notions of uncertainty measurements defined for NRP. According to the experiments upon which this paper reports, the proposed two robust MONRP approaches (*MCNRP-US* and *MCNRP-R*) overcome the limitation of the traditional MONRP approach which underestimates (or even hides) uncertainty. These allow the decision maker to choose different approaches for controlling different types of uncertainty, while retaining the performance of traditional solutions. The *MCNRP-US* offers decision makers a way to control the fluctuation range of *payoff* for solutions. The *MCNRP-R* model helps decision makers to explore solutions with lower *risk* of budget overrun.

We found that MONRP-R decisions are more closely correlated to traditional MONRP decisions regarding requirement choice, than MONRP-US. We also find that, while cost is closely correlated to inclusion of a requirement in the Pareto-front, revenue is not.

In this paper, only triangle probability distribution uncertainty has been examined. Real world requirement engineering analysis and optimization problems typically contain a mixture of types of uncertainties. Hence, future work will focus on adapting and evaluating our approach on other real world scenarios.

9. REFERENCES

[1] A. Al-Emran, P. Kapur, D. Pfahl, and G. Ruhe. Studying the impact of uncertainty in operational release planning - an integrated method and its initial evaluation. *Information and Software Technology*, 52(4):446–461, Apr. 2010.

[2] A. Al-Emran, D. Pfahl, and G. Ruhe. Decision support for product release planning based on robustness analysis. In *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference (RE'10)*, pages 157–166, Washington, DC, USA, 2010. IEEE Computer Society.

[3] A. J. Bagnall, V. J. Rayward-Smith, and I. M. Whittley. The next release problem. *Information and Software Technology*, 43(14):883–890, 2001.

[4] P. Baker, M. Harman, K. Steinhofel, and A. Skaliotis. Search based approaches to component selection and prioritization for the next release problem. In *Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM'06)*, pages 176–185, Washington, DC, USA, 2006. IEEE Computer Society.

[5] D. Bertsimas and M. Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[6] H.-G. Beyer and B. Sendhoff. Robust optimization–a comprehensive survey. *Computer Methods in Aplied Mechanics and Engineering*, 196(33):3190–3218, 2007.

[7] B. W. Boehm. *Software Engineering Economics.* Prentice Hall, Upper Saddle River, NJ, USA, 1st edition, 1981.

[8] A. Budzier. Why your it project may be riskier than you think. *Harvard Business Review*, 89(9):23–25, 2011.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Transaction on Evoluionary Computation*, 6(2):182–197, Apr 2002.

[10] J. C. Felli and G. B. Hazen. Sensitivity analysis and the expected value of perfect information. *Medical Decision Making*, 18(1):95–109, 1998.

[11] 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*, pages 115–124, Sept 2008.

[12] D. Greer and G. Ruhe. Software release planning: an evolutionary and iterative approach. *Information and Software Technology*, 46(4):243–253, 2004.

[13] J. M. Hammersley, D. C. Handscomb, and G. Weiss. Monte carlo methods. *Physics Today*, 18:55, 1965.

[14] M. Harman, I. M.-B. Jens Krinke, J. R. Francisco Palomo-Lozano, and S. Yoo. Exact scalable sensitivity analysis for the next release problem. In *ACM Transactions on Software Engineering and Methodology, 2014.* ACM, 2014. To appear.

[15] M. Harman and B. F. Jones. Search-based software engineering. *Information and Software Technology*, 43(14):833–839, 2001.

[16] 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)*, pages 1681–1688, New York, NY, USA, 2009. ACM.

[17] E. Letier, D. Stefan, and E. T. Barr. Uncertainty, risk, and information value in software requirements and architecture. *International Conference on Software Engineering*, 2014.

[18] 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)*, pages 771–778, New York, NY, USA, 2005. ACM.

[19] G. L. Nemhauser and Z. Ullmann. Discrete dynamic programming and capital allocation. *Management Science*, 15(9):494–505, 1969.

[20] 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)*, pages 1469–1476, New York, NY, USA, 2013. ACM.

[21] A. Saltelli, S. Tarantola, and F. Campolongo. Sensitivity analysis as an ingredient of modeling. *Statistical Science*, 15(4):377–395, 2000.

[22] Y. Zhang, E. Alba, J. J. Durillo, S. Eldh, and M. Harman. Today/future importance analysis. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO'10)*, pages 1357–1364, New York, NY, USA, 2010. ACM.

[23] 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)*, pages 1129–1137, New York, NY, USA, 2007. ACM.

[24] Z. Zi. Sensitivity analysis approaches applied to systems biology models. *Systems Biology, IET*, 5(6):336–346, Nov 2011.

[25] 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.