Search Based Data Sensitivity Analysis Applied to Requirement Engineering

Mark Harman, Jens Krinke, Jian Ren and Shin Yoo
King's College London, Centre for Research on Evolution, Search and Testing
Strand, London
WC2R 2LS, UK

ABSTRACT

Software engineering is plagued by problems associated with unreliable cost estimates. This paper introduces an approach to sensitivity analysis for requirements engineering. It uses Search-Based Software Engineering to aid the decision maker to explore sensitivity of the cost estimates of requirements for the Next Release Problem (NRP). The paper presents both single- and multi-objective formulation of NRP with empirical sensitivity analysis on synthetic and real-world data. The results show strong correlation between the level of inaccuracy and the impact on the selection of requirements, as well as between the cost of requirements and the impact, which is as intuitively expected. However, there also exist a few sensitive exceptions to these trends; the paper uses a heat-map style visualisation to reveal these exceptions which require careful consideration. The paper also shows that such unusually sensitivity patterns occur in real-world data and how the proposed approach clearly identifies them.

Categories and Subject Descriptors

D.2.1 [**SOFTWARE ENGINEERING**]: Requirements/ Specifications—*Methodologies*

General Terms

Algorithms, Measurement, Experimentation.

1. INTRODUCTION

One of the common problems in requirements engineering is caused by the uncertainties that are inherent in the decisions made by the requirement engineer. Most of the data needed by the requirement engineer, such as expected revenue, development cost or duration, is inherently unknown at the time of requirement planning stage. The clients of the product also contribute to these uncertainties because often they do not possess clear idea about which features they want before actually see it. Naturally, the requirement engineer has to balance many complex criteria based on estimated data.

It is a well-known fact that cost estimation is a difficult and demanding activity [1, 6]. It is also widely believed that the cost estimates produced by software engineers tend to include a great degree of inaccuracy [10, 11]. This is not due to the ineptness of the requirement engineer; it is rather because of the astonishingly wide variety in the problems that software engineering faces. Unlike other engineering disciplines, there are fewer well-understood construction approaches.

This paper does not attempt to resolve the inaccurate cost estimate problem; it seems that the problem will remain unsolved for the foreseeable future of software engineering. Rather, the paper seeks to introduce an approach to provide the requirements engineer with a decision support system guided by Search-Based Software Engineering (SBSE). The approach assesses the impact of inaccuracies of the cost estimation of each requirement on the solutions to the requirements allocation problem, known as the Next Release Problem [2]. The Next Release Problem is the problem of selecting the software requirements to be implemented in the next release of a product so that benefits such as customer satisfaction or revenue are maximised while all the constraints such as budget are satisfied. The decision support system aids the requirement engineer by identifying the sensitive regions in the estimated data which will lead to relatively higher impact on the selection of the requirements. This information then can be used to focus the effort on obtaining more accurate estimation of those requirements.

Each set of estimates and customer choices denotes a separate and unique optimisation problem. The structure of the data and the relationships between estimated data may create unexpected interactions between requirement estimates, which can yield a butterfly effect; a small inaccuracy in a low cost requirement can have an unexpectedly large effect on the overall decision. Because of the size of the data sets involved and the inherent complexity of the interactions between estimates, it is nearly impossible for an engineer to fully comprehend these hidden relationships without automated decision support.

The intuitive answer to the sensitivity of cost estimation problem is that the more expensive the requirement is, the greater impact it will have on the result when estimated inaccurately. Also, similarly, it can be expected that the higher the level of inaccuracy, the bigger the impact it has. The paper indeed statistically confirms these intuitive assumptions. However, the paper also reports that there are exceptions to these trends. It is these exceptions that require careful cost estimation, because they can have unexpectedly high impact on the selection of requirements. The paper uses a heat-map style visualisation, generated using a search-based approach, to identify these sensitive exceptions in the data. The hot-spots on the heat map will indicate areas where a particular inaccuracy level for a particular requirement estimate can lead to high impact. The heat-map provides an intuitive visual aid to comprehend the complex interaction in the data set.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

The paper presents two different formulations of the problem. With the single-objective formulation, the requirements engineer assesses the impact of inaccuracy at a specific level on weighted customer satisfaction values. In this model, the requirements engineer knows the expected inaccuracy and seeks to identify overall budget levels and particular requirements that are sensitive to this. The second formulation is the multi-objective formulation in which the requirements engineer simply seeks to reduce estimated cost and increase estimated revenue, but does not know how inaccurate the estimates are likely to be. The single-objective formulation is more appropriate for a mature organisation with a history of development and a consequent knowledge of likely levels of estimate inaccuracy. The multi-objective formulation has the advantage that it can be applied without any knowledge of likely estimate inaccuracy levels.

Both formulations are applied to both synthetic and real world data. The primary contributions of the paper are as follows:

1. The paper shows how SBSE can be used as a technique for sensitivity analysis in requirements engineering.
2. The paper presents two formulations of the NRP of requirements engineering and shows how SBSE can be used for both formulations, presenting an evaluation using real world data and synthetic data.
3. The paper shows how heat-maps can be used to intuitively identify unexpectedly sensitive requirements estimates to guide the decision maker, providing insight into the structure of their estimate data.

2. BACKGROUND

The paper presents a sensitivity analysis for two different formulations of the Next Release Problem (NRP): single-objective version and multi-objective version.

2.1 Single-objective Next Release Problem

The single-objective formulation follows the definition of NRP by Bagnall et al. [2]. First, it is assumed that for an existing software system there is a set of possible software requirements, denoted by:

$$\mathcal{R} = \{r_1, \dots, r_n\}$$

For the sake of simplicity, it is also assumed that there is no dependency relation between those requirements. Bagnall et al. note that any instance of NRP with dependency relation can be converted to a *basic* NRP by merging the requirements that belong to dependency chains [2].

The cost of fulfilling this set of requirements r_i ($1 \leq i \leq n$) is denoted by:

$$Cost = \{cost_1, \dots, cost_n\}$$

The expected revenue of every possible requirement is denoted by:

$$Revenue = \{revenue_1, \dots, revenue_n\}$$

The decision problem form of NRP is the question of finding the optimal subset(s) of requirements to maximise the total revenue and minimise the cost of development.

The decision vector, $\overrightarrow{X}$, is represented by:

$$\overrightarrow{X} = < x_1, \dots, x_n >$$

where the ith element of $\overrightarrow{X}$ is 1 if the ith requirement is to be implemented and 0 if it is not. Now, given an instance of the decision vector, $\overrightarrow{X_1}$, its fitness, $F(\overrightarrow{X_1})$, is the sum of expected revenues for the requirements to be implemented by $\overrightarrow{X_1}$:

$$F(\overrightarrow{X_1}) = \sum_{i=1}^{n} revenue_i \cdot x_i$$

Similarly, the cost of implementing a set of requirements represented by $\overrightarrow{X_1}$ is:

$$cost(\overrightarrow{X_1}) = \sum_{i=1}^{n} cost_i \cdot x_i$$

Given a budget of b, the single-objective NRP is a problem of finding a decision vector $\overrightarrow{X}$ such that $F(\overrightarrow{X})$ is maximised while satisfying $cost(\overrightarrow{X}) \leq b$:

$$\text{Maximise } \sum_{i=1}^{n} revenue_i \cdot x_i$$

$$\text{while subject to } \sum_{i=1}^{n} cost_i \cdot x_i \leq b$$

2.2 Multi-objective Next Release Problem

The multi-objective Next Release Problem (MONRP) is a multi-objective optimisation version of NRP. In multi-objective optimisation problems, there are multiple objectives expressed in fitness functions, which are often in conflict with each other [14]. In case of MONRP, it can be said that the expected revenue and the development cost of a product are in conflict with each other.

The multi-objective formulation is defined following Zhang et al. [18]. Unlike the single-objective formulation, the cost is no longer a constraint. In multi-objective formulation, the development cost is minimised while the expected revenue is maximised.

$$\text{Maximise } \sum_{i=1}^{n} revenue_i \cdot x_i, \quad and$$

$$\text{Minimise } \sum_{i=1}^{n} cost_i \cdot x_i$$

In multi-objective optimisation, a solution A is said to dominate a solution B if and only if A is at least equal to B in all objectives, and excels B in at least one objective. This is called Pareto-optimality. As a result, a solution of a multi-objective optimisation problem is expressed in a Pareto-front, which is a set of multiple solutions that do not dominate each other.

3. SENSITIVITY ANALYSIS IN NRP

Since the models used in the empirical studies are small enough to be solved quickly, a *brute force* approach is implemented for sensitivity analysis: simply modify the initial input data and run the algorithm repeatedly to see how the result changes. Figure 1 illustrates the difference between the general optimisation process and sensitivity analysis. The cost of each requirement is modified to simulate the inaccurate estimation. The data are then fed into a meta-heuristic optimisation algorithm designed for NRP, which will produce an alternative solution. The impact is then evaluated by measuring the distance between the original solution and the alternative solution.

There are two critical elements that are required in order to simulate *what-if* scenarios in which a particular estimation is inaccurate. First, the algorithm used to solve NRP has to be deterministic; otherwise it is impossible to determine whether the observed change in the result is due to the inaccurate estimation or the randomness of the algorithm.

In most (if not all) of multi-objective evolutionary algorithms, *Pseudo Random Numbers(PRNs)* are used in the procedure of evolutionary calculation. For instance, pseudo random numbers are

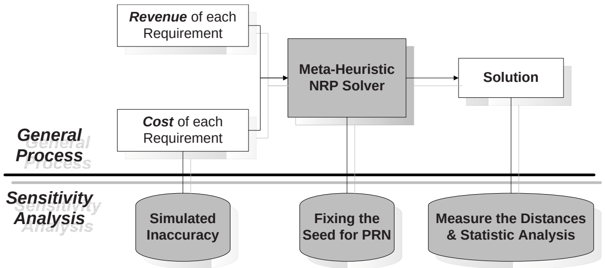

Figure 1: Sensitivity Analysis Flow Chart

used in generating the initial population, selecting the bits in candidates to perform mutation and crossover. Due to the inherent randomness of the evolutionary algorithms, and the fact that it cannot guarantee the global optimum, every 'run' of the implementation would provide a different result even if the input data are identical. In order to perform the sensitivity analysis, we need to distinguish the difference between the indeterminacy of the algorithm itself and the changes caused by the modification on the input data.

We introduce a fixed seed for *PRN* to provide an identical sequence of PRNs for each execution of the implementation. Because the problem definition and the problem model are not changed during the optimisation process, given the identical sequence of PRN, we ensure that the change on the result is caused by the change on the input data only.

The second element required by sensitivity analysis is a method that can measure the changes brought in by the error in quantitative manner. If it is not possible to express the changes in quantitative forms, it would also be impossible to compare the criticality of errors. The actual method of measurement is specific to the definition and representation of the problem.

With the single-objective formulation, we evaluate the difference between two decision vectors by their Hamming distance. This is possible because the greedy algorithm, which we use to solve the single-objective NRP, produces a single solution to an instance of NRP problem. However, the NSGA-II, which is the algorithm we use to solve the multi-objective NRP, does not produce a single solution, but a set of solutions that form Pareto Frontier. Therefore, the difference should be measured between two sets of solutions (two optimal fronts), not two different solutions. In order to measure the distance between two sets of solutions, the Generation Distance is used [17]. It is based on the calculations of *Euclidean Distance* of the solutions between two fronts.

To calculate the distance between two fronts (f_a, f_b) of two different executions of optimisation, we define (A_1, A_2, ..., A_n) to denote the n solutions belonging to front f_a while (B_1, B_2, ..., B_m) denote the m solutions belonging to front f_b, where n and m are the numbers of solutions contained by each front respectively.

The distance from solution A to solution B is the Euclidean distance between objective values normalised to [0,1]. In the case of two-objectives NRP, the distance between A and B is defined as:

$$Dis(A, B) = \pm\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

where, in our case, x_a and x_b is the normalised overall cost for solution A and B respectively while y_a and y_b are the normalised revenues.

The distance from one particular point A to f_b is considered as the same distance from A to its geometrically closest point on front f_b. Distance between A and front f_b is defined as:

$$Dis(A, f_b) = Dis(A, B)$$

where B is the closest point to A on front f_b.

The distance from front f_a to f_b is then calculated as the mean value of the distance from every point on f_a to f_b.

$$Dis^*(f_a, f_b) = \frac{\sum_{i=1}^{n} Dis(A_i, f_b)}{n}$$

where n is the number of optimal solutions on front f_a.

Finally, in order to achieve fair contributions from both fronts to the distance calculation, we develop the formulation to calculate the distance between two Pareto fronts f_a and f_b as below:

$$Distance(f_a, f_b) = \frac{Dis^*(f_a, f_b) + Dis^*(f_b, f_a)}{2}$$

4. EXPERIMENTAL SETUP

The argument of the data sensitivity problem is based on the assumption that some of the estimated quantitative data may contain some error. The amount of the actual error will be known only afterwards. However, it is possible to measure the repercussions of the potential errors by trying out various *what-if* scenarios. If an introduction of a certain deliberate error to a specific part of data creates large amount of change in the final solution, it would be safe to say that the specific part of data is highly sensitive to an error. With this knowledge, the decision maker can manage the potential risks more efficiently, as well as concentrate on elaborating the estimation of more sensitive data.

In case of NRP, the most important scenarios are the cases when the costs of some requirements are based on wrong estimation. The decision maker would want to know which requirement will create the most significant change in the final solution if there is an error in the estimation of its development cost. Therefore, the scenarios in this case will be different versions of the data, each containing a requirement with modified development cost. The alternative solution will be a subset of requirements selected based on the modified data. If the alternative solution is radically different from the original solution, it indicates that the introduced error brings in a significant change. If this process is repeated for each requirement with the same margin of error, it is possible to identify the requirement that is most sensitive to the same level of inaccuracy.

The intuitive answer to the cost sensitivity analysis problem is that the more expensive a requirement is, the bigger impact it will have if its cost is estimated inaccurately. Similarly, it can be said that the more inaccurate the estimation is, the bigger impact it will have on the result of NRP. We hereby call this the *Positive Correlation Assumption* (PCA). More specifically, we denote the first assumption (between cost and impact) by PCA-1, and the second assumption (between inaccuracy and impact) by PCA-2. These assumptions are statistically tested against both synthetic and real-world requirement data. For this, the empirical studies utilise the greedy algorithm and NSGA-II to single- and multi-objective formulations of NRP with deliberate errors in the data set.

4.1 Greedy Algorithm

The greedy algorithm is known to be efficient and effective for 0-1 knapsack problem, which is the basis of NRP. It is constructive in nature and start with an empty set of selected requirement. At each iteration, a requirement is added to the set until no further additions can be made without exceeding the given budget. The choice of which requirement to select at each iteration is guided by the fitness value.

First of all, all the requirements are sorted by their revenue in descending order and then by cost in ascending order if their revenues are the same. All those requirements at the front of the queue will be then selected into the solution vector until the budget bound has been reached. Algorithm 1 shows the pseudo-code of the greedy algorithm used in the paper.

```
input  : N:number of requirements; cost; budget
output: solution; currentCost
Sort the requirements in the order of descending revenue
and then in the order of ascending cost if they share the
same revenue
for i ← 1 to N do
    if currentCost + cost(i) ≤ budget then
        currentCost ← currentCost + cost(i);
        solution(i) ← 1;
    end
end
```

Algorithm 1: Greedy Algorithm

4.2 NSGA-II

The recent implementation of NSGA-II [9] from Zhang et al. [18] for multi-objective NRP is used in a simplified version. Initially, a random parent population P_0 is created. The population size is N. The population is sorted using the non-dominated relations. Each solution is assigned a fitness value equal to its non-domination level. Binary tournament selection, crossover, and mutation operators are used to create a child population Q_0 of size N. Then the NSGA-II procedure goes to the main loop which is described in Algorithm 2. Maximising the overall revenue and minimising the overall cost of each solution are considered as the two objectives for NSGA-II.

```
while not stopping rule do
    Let R_t = P_t ∪ Q_t;
    Let F = fast-non-dominated-sort(R_t);
    Let P_{t+1} = ∅ and i = 1;
    while |P_{t+1}| + |F_i| ⩽ N do
        Apply crowding-distance-assignment(F_i);
        Let P_{t+1} = P_{t+1} ∪ F_i;
        Let i = i + 1;
    end
    Sort(F_i, ≺ n);
    Let P_{t+1} = P_{t+1} ∪ F_i[1 : (N-|P_{t+1}|)];
    Let Q_{t+1} = make-new-pop(P_{t+1});
    Let t = t + 1;
end
```

Algorithm 2: NSGA-II Algorithm

4.3 Requirement Data

The paper uses two sets of synthetically generated data as well as a set of real-world requirements data obtained from a large telecommunication company. The first synthetic data is generated purely randomly, i.e. there is no correlation between the cost of a requirement and its expected revenue, which is connected to its fitness value in the optimisation problem. The second set is generated so that the cost of a requirement has a positive correlation with its expected revenue. Each of the two sets of synthetic data contains 30 requirements. The cost and revenue for each requirement are generated using the uniform distribution over the interval of [1, 1500] and [1, 10] respectively. Comparing the results from these two synthetic data set allows us to test the statistical significance of PCA.

The real-world requirement data is obtained from Motorola. It originally contains 40 different features that are interference-free, i.e. any combination of which can be implemented into a single product. However, 5 features that represent the core functionality of the product were combined by dependencies between themselves, and it was decided that they will always be included in the final selection of requirements. This left us 35 features with so sparse a dependency relationship that it could be ignored.

4.4 Evaluation

We modify the cost of each requirement only one at a time using 21 different Percentage Increase in Actual Cost (PIAC) values ranging from -50% to 50% with steps of 5%. A positive PIAC value means that the actual cost has increased compared to the estimated cost, which means an underestimation; a negative PIAC value means that the actual cost has decreased compared to the estimated cost, which means an overestimation. The Hamming distance and the Euclidean distance between the results from the modified data and the original data is used to quantify the difference observed in the multiple executions. Spearman's rank correlation coefficient is used to test PCA and analyse how the changes on result relates to the modifications of initial data.

4.5 Research Questions

The paper presents the following research questions. **RQ1** and **RQ2** concern the statistical significance of PCA.

RQ1: Does the sensitivity analysis confirm PCA-1, i.e. the correlation between the cost of a requirement and its impact on NRP with statistical significance?

RQ2: Does the sensitivity analysis confirm PCA-2, i.e. the correlation between the level of inaccuracy and its impact on NRP with statistical significance?

RQ1 and **RQ2** is quantitatively answered using Spearman's rank correlation analysis in Section 5. The third research question inherently requires qualitative analysis.

RQ3: Is there any exception to the general trend observed by PCA?

RQ3 is answered by analysing the heat-map visualisation in Section 5.

5. RESULTS AND ANALYSIS

5.1 Result From Single-Objective Formulation

Figure 2[1] shows four heat-map visualisations from the results of sensitivity analysis on Motorola's data set, using single-objective formulation of NRP. The x-axis corresponds to different instances of NRP, sorted in ascending order of the budget assigned to each instance. The y-axis corresponds to different requirements, sorted in ascending order of their estimated cost. The two heat-maps on the left show the Hamming distance between the original greedy algorithm solutions and the alternative solutions with PIAC value of $\pm 25\%$, i.e. the underestimate or overestimate error by 25% margin. Similarly, the two on the right show the results with PIAC value of $\pm 50\%$, i.e. the underestimate overestimate error by 50% margin. A darker colour represents a bigger Hamming distance.

The heat-map reveals the complex interaction between the budget and the revenue and cost of each requirement. A single requirement shows varying levels of sensitiveness depending on the combination of the budget and the margin of error. However, some straightforward patterns can be easily observed. First, errors on expensive requirements do not have any impact on smaller budgets if the original estimated cost and the modified cost are both larger than the given budget, of which the fact is reflected by the white area in the left lower corner of all four heat-maps. Second, when comparing the PIAC value of four heat-maps, the bigger PIAC

[1] The animated GIF that depicts the evolution of sensitivity as budget level could be found at: http://www.dcs.kcl.ac.uk/pg/renjian/gecco2009.gif.

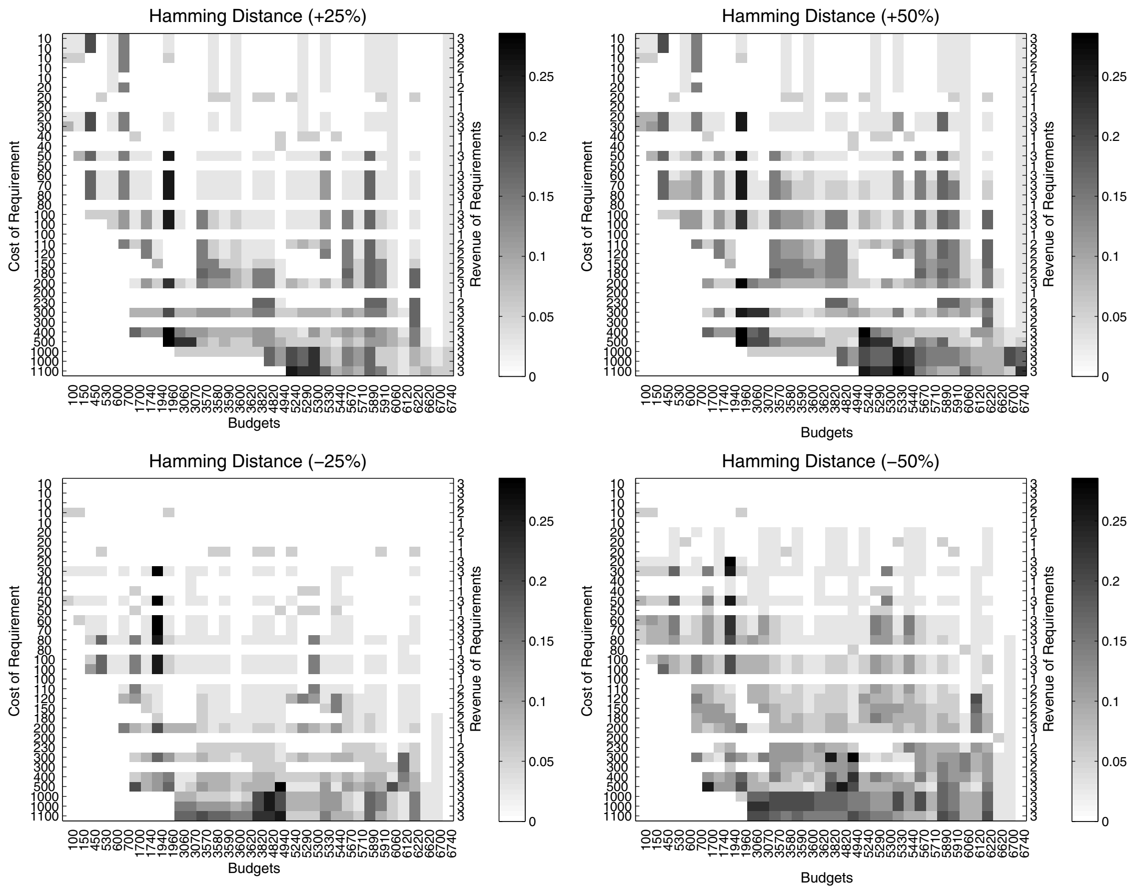

Figure 2: Hamming distance from the original solution to the solution obtained by the greedy algorithm with PIAC value of ±25% and ±50%.

value tend to bring more impact on the results. Third, when comparing the cost of each requirement, more expensive requirements tend to have bigger impact on the results. On the other hand, some cheaper requirements do not have any impact on the result if and only if their cost is overestimated (PIAC = −25%, −50%), in which cases the amount of errors is too small to free enough space on given budget for a more expensive requirement to be filled in. The last interesting observation is that some expensive requirements but with low revenue do not have any impact on Hamming distance across all budget values, since these requirements are sorted into the tail of the waiting queue, where the requirements are not likely to be selected.

However, due to the existence of budget constraint, it is not possible to visualise the trend with respect to the cost, the expected revenue, and the PIAC value at the same time. For this, we turn to the multi-objective formulation; since the Euclidean distance between two fronts incorporate differences in both the revenue and the cost, we can observe the trend between PIAC and its impact.

5.2 Result From Multi-Objective Formulation

Figure 3 shows the heat-map visualisation generated from the sensitivity analysis for the MONRP formulation. The x-axis represents different PIAC values, ranging from −50% (overestimation) to 50% (underestimation). The y-axis represents different requirements, sorted by their development cost. By cross-referencing x-axis and y-axis, it is possible to observe how much impact it makes to underestimate or overestimate the cost of a specific requirement by the given degree of error. The darker the colour is, the bigger impact the particular error has.

A few trends can be easily observed. First, one of the dominant trends across all three data sets is that the distance between the original and inaccurate front increases as PIAC value increases. Second, when comparing the cost of those requirements, more expensive requirements tend to have bigger impact on the results. These two observations are statistically tested in Section 5.3.

However, there are a few exceptions to the general trend. Certain requirement almost consistently has significant impact on the result. For example, the second requirement in the real-world data set consistently produces a noticeable Euclidean distance from PIAC value of −5% to −50%. This consistency provides two interesting insights into the real-world data set. First, this particular requirement brings about significant impact on the result even when its cost is reduced only by 5% (PIAC = −5%). Second, and more interestingly, further reduction in its cost still produces the same level of impact up to reduction of 50% (PIAC = −50%). This is due to the fact that the particular requirement has the lowest cost and lowest expected revenue among the requirements. It is possible to conclude that the threshold for overestimation of this particular requirement is 5%. If the PIAC value reaches the threshold value, the final solution will be different from the original solution.

5.3 Statistical Analysis

Figure 4 and Figure 5 show the boxplots of Euclidean distances measured with different sets of data. Each boxplot in Figure 4 represents the Euclidean distances measured from all requirements that share the same value of development cost. Each boxplot in Figure 5 represents the Euclidean distances measured from all requirements in the data set for a specific PIAC value. In both fig-

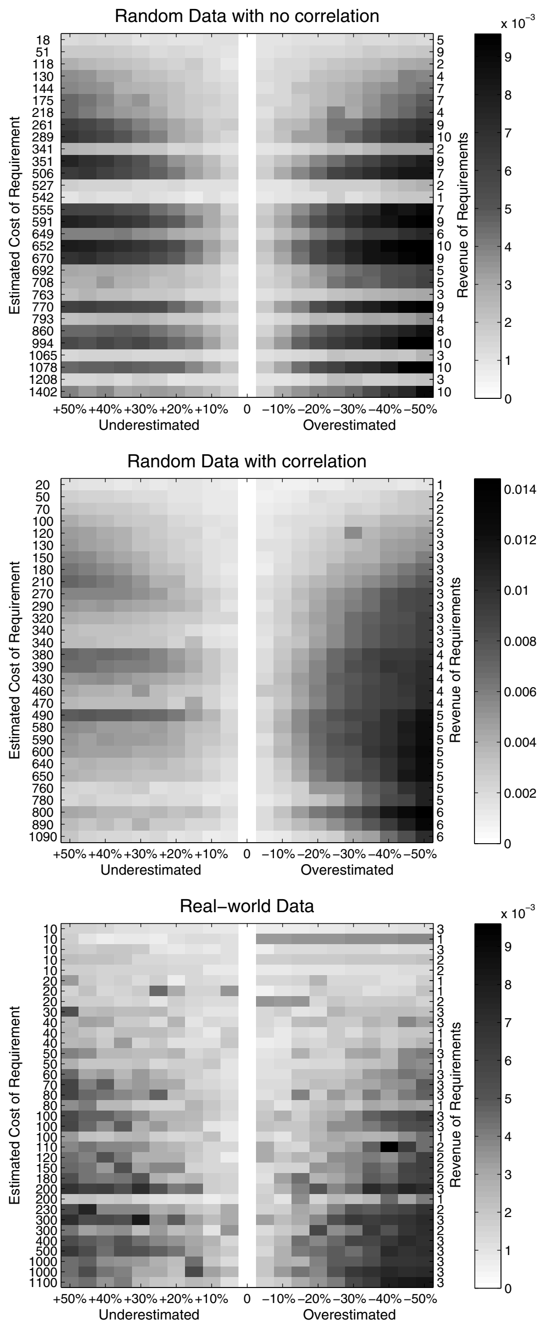

Figure 3: Euclidean distance between original estimated Pareto-front and actual Pareto-front by different PIAC values.

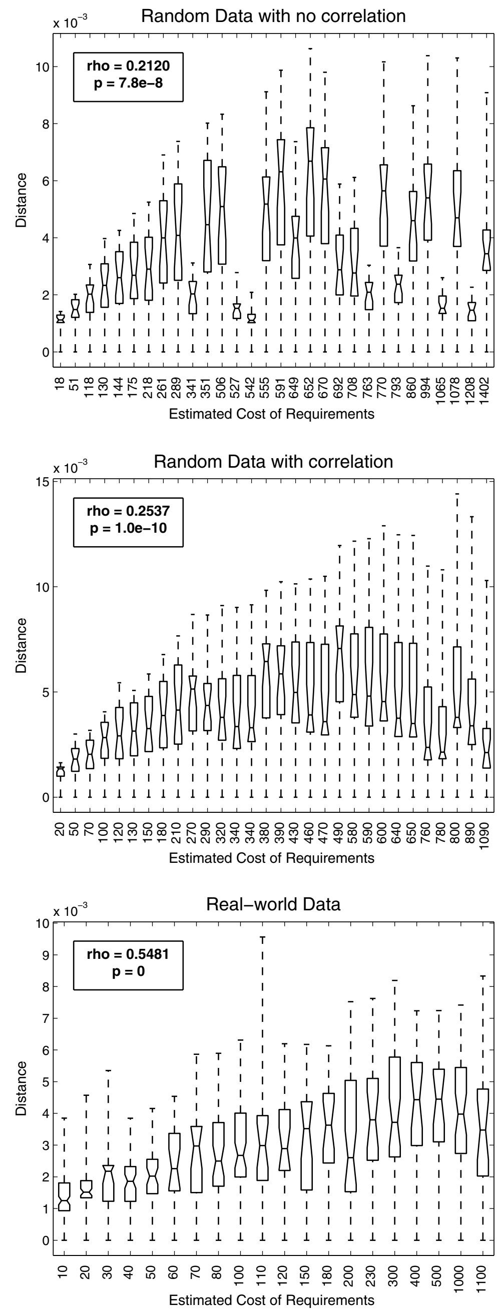

Figure 4: Boxplots of Euclidean distances between Pareto-fronts for different costs of requirements

ures, the general trend is a positive correlation between Euclidean distance and PIAC or cost, meaning that larger PIAC values and larger development cost will have greater impact on the result.

The random data set with no correlation between cost and revenue shows a several unique data points that do not follow the overall trend. The position and number of these exception correspond to the exceptions observed in the corresponding heat-map in Figure 3. This implies that, if the data set contains requirements that do not fit the Positive Correlation Assumption, there are likely to exist exceptional requirements. With the random data set with positive correlation between cost and revenue, the PCA trend is more consistent and smooth.

To test PCA statistically, Spearman's rank correlation coefficient ρ is used to quantitatively describe the relationship between two pairs of separate variables. On the other hand, the p-value, which is the result of the permutation test, indicates whether the calculated value of ρ is significant to prove that there is a monotonic relationship between the pair of variables. The smaller the p-value is, the stronger monotonic correlation between the pair of compared variables exists. A p-value of 0.05 indicates that 95 times in 100 the monotonic relationship between two sets of variables occurred because a correlation exists, and not because of pure chances.

Figure 4 and Figure 5 are statistically analysed using Spearman's rank correlation analysis. The ρ (rho) values and the corresponding

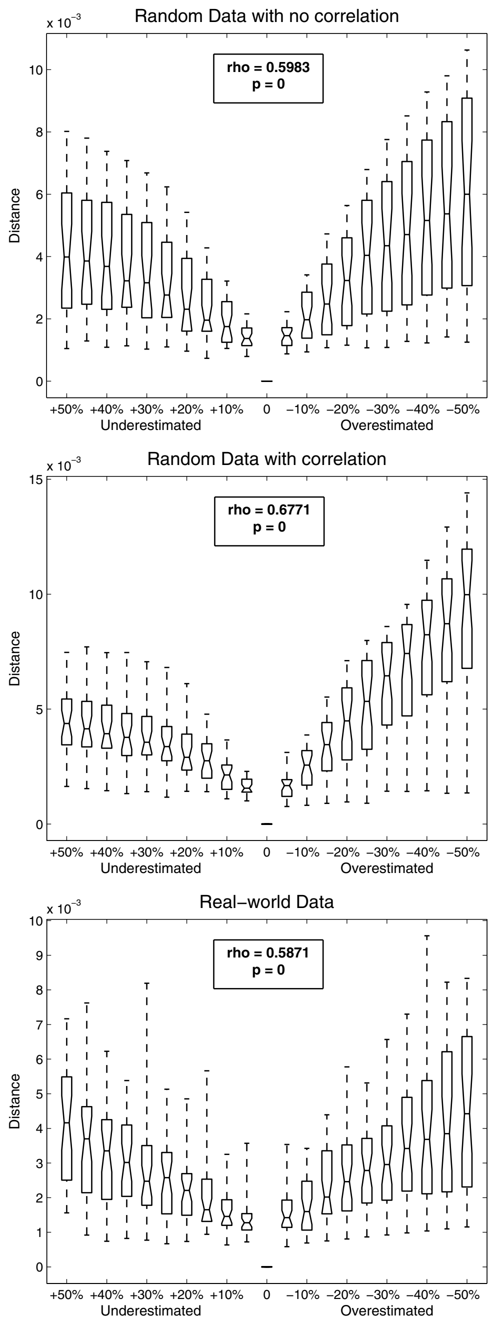

Figure 5: **Boxplots of Euclidean distances between Pareto-fronts for different PIAC values**

p-values are calculated and plotted in each figure. All the p-values are very close to zero,which indicates strong monotonic correlation between each pair of variables, meaning that larger PIAC values and larger development cost will have greater impact on the result. The visual observations of general trend of PCA-1 and PCA-2 are confirmed.

Table 1 shows the Spearman's correlation coefficient and the p-values calculated for the relation between cost of requirements and Euclidean distance between two Pareto fronts for Motorola's data set on each level of PIAC. All the p-values are smaller than 0.05 which indicates the strong monotonic correlation between cost and impact. Again, this confirms the general trend predicted by PCA-1 at each level of PIAC.

PIAC	ρ_{cost}	p_{cost}	*PIAC*	ρ_{cost}	p_{cost}
+50%	0.6648	0.0000	−50%	0.9246	0.0000
+45%	0.8204	0.0000	−45%	0.8918	0.0000
+40%	0.7386	0.0000	−40%	0.8912	0.0000
+35%	0.8093	0.0000	−35%	0.8940	0.0000
+30%	0.7104	0.0000	−30%	0.8646	0.0000
+25%	0.6194	0.0001	−25%	0.8520	0.0000
+20%	0.5704	0.0003	−20%	0.7560	0.0000
+15%	0.8155	0.0000	−15%	0.6370	0.0000
+10%	0.6256	0.0001	−10%	0.4645	0.0049
+5%	0.4239	0.0112	−5%	0.3600	0.0336
0%	0.5026	0.0021			

Table 1: **Spearman's rank correlation coefficient and p-values between cost and impact. For all PIAC values, the ρ values are statistically significant at the confidence level of 95%.**

Req.	ρ_{PIAC}	p	*Req.*	ρ_{PIAC}	p
1	0.9474	0.0000	19	0.7318	0.0002
2	0.8591	0.0000	20	0.8929	0.0000
3	0.8825	0.0000	21	0.7188	0.0002
4	0.8591	0.0000	22	0.8591	0.0000
5	0.9604	0.0000	23	0.7786	0.0000
6	0.9630	0.0000	24	0.7890	0.0000
7	0.7942	0.0000	25	0.8721	0.0000
8	0.7890	0.0000	26	0.9136	0.0000
9	0.9032	0.0000	27	0.6929	0.0005
10	0.4487	0.0413	28	0.8617	0.0000
11	0.8643	0.0000	29	0.2071	0.3676
12	0.8773	0.0000	30	0.8877	0.0000
13	0.9266	0.0000	31	0.9578	0.0000
14	0.3188	0.1589	32	0.8123	0.0000
15	0.6305	0.0022	33	0.9162	0.0000
16	0.8981	0.0000	34	0.8331	0.0000
17	0.2461	0.2822	35	0.7838	0.0000
18	0.6929	0.0005			

Table 2: **Spearman's rank correlation coefficient between PIAC value and impact. For most requirements, the observed ρ values are statistically significant at the confidence level of 95%.**

Similarly, Table 2 shows the ρ and the p-value for the PIAC values and Euclidean distance for each requirement from Motorola's data set. Again, the result confirms the general trend predicted by PCA-2.

5.4 Answers to the Research Questions

RQ1 and **RQ2** are answered by the statistical analysis shown in Table 1 and Table 2. The Spearman's rank correlation coefficient confirms that there exists a positive correlation between the cost of each requirement and the impact, and between the level of inaccuracy and the impact. The correlation is statistically significant with confidence level of 95%.

However, it is the overall trends and the exceptions observed in Figure 3, Figure 4 and Figure 5 that would be of particular interest to the decision maker. First, while the PCA is statistically confirmed in general, there are exceptions to the trends. In Figure 3, the heat-map for the random data set with no correlation shows that the requirements that have relatively high and low impact factor from distinct horizontal bands. This phenomenon is weakened in the second heat-map for the random data set with correlation. Finally, the real world data shows much more complex patterns with very few distinct horizontal bands.

Comparing the first and the second heat-map, it can be said that the correlation between the cost and the expected revenue of requirements is an important factor in sensitivity analysis. More specifically, if it is likely that some requirements have high cost and low revenue, or vice versa, these requirements are more likely to contribute to create the sensitive region in NRP solution.

Figure 4 and Figure 5 also visually confirm PCA-1 and PCA-2 respectively. In Figure 4, we can observe unique boxplots with very small variance which correspond to the low-impact horizontal bands observed in the first heat-map in Figure 3. Another interesting observation found in Figure 5 is that overestimation tends to have bigger impact on the solutions of NRP than underestimation; boxplots on the right side of Figure 5 shows steeper increase in mean values than those on the left side. This trend has a very interesting implication to practitioners, because under uncertainties, a human decision maker is more likely to overestimate than underestimate. This qualitative assessment of the statistical analysis forms the answer to **RQ3**.

6. RELATED WORK

In the Next Release Problem (NRP), the goal is select an optimal subset of requirements for the next release of a product. Bagnall et al. first suggested the term NRP and applied various modern heuristics including greedy, hill climbers and simulated annealing algorithm [2]. Baker et al. [3] applied Search-Based Software Engineering approach to NRP by using single objective optimisation algorithms: the greedy algorithm and the simulated annealing algorithm. A variation of the problem using integer linear programming is studied in Van den Akker's work [16], to find exact solutions within budgetary constraints.

Zhang et al. [18] introduced new formulations of Multi-objective Next Release Problem(MONRP). In Zhang's MO-NRP formulations, at least two parameters (possibly conflicting) are considered as two optimisation objectives simultaneously.

Sensitivity analysis has been widely applied in various areas including complex engineering system, environmental studies, economics, health care, etc. [4, 7, 12, 13] It has been used as one of the principal quantitative techniques in risk management [5]. It can be used to provide an insight of the reliability and robustness of a problem model result when making decisions [15]. However, the present paper is the first to introduce Sensitivity Analysis in multi-objective optimisation problem in the area of software engineering.

The proof-of-principle study by Deb et al. [8] introduced robust optimisation procedures to multi-objective optimisation problems for the purpose of searching for robust Pareto-optimal solutions in multi-objective optimisation problems. It is worth mentioning that robust optimisation is concerned with finding areas of the solution space that change little. Our approach to sensitivity analysis is concerned with measuring the impact on the solution proposed to changes (small and large) in the components that serve to make up a candidate solutions. Whereas robustness helps the decision maker to choose a good solution, sensitivity analysis helps the decision maker to re-focuss estimation effort on the problem description data that most requires careful estimation.

7. CONCLUSIONS AND FUTURE WORK

The paper introduces an SBSE approach to identify requirements that are anomaly sensitive to inaccurate cost estimation. Sensitive requirements are those that have significant impact on the final solution of NRP when their cost estimates are inaccurate. The paper presents an automated sensitivity analysis approach based on SBSE for both single- and multi-objective NRP formulations. The results of the sensitivity analysis is summarised in an intuitive heat-map style visualisation to aid the decision maker to identify sensitive regions in the data.

Through the empirical studies of both synthetic and real-world requirement data, the paper presents a statistical analysis that confirms the Positive Correlation Assumption, i.e. more expensive requirements and higher level of inaccuracies tend to have greater impact on NRP. However, the heat-map visualisation also reveals that there exist exceptions to this assumption. Identifying these exceptions can guide the decision maker towards more accurate estimation and safer decision making. Future work will consider more complex aspects of NRP such as complex dependency relation between requirements.

8. REFERENCES

[1] Armour, P. Ten Unmyths of Project Estimation. *Communications of the ACM 45*, 11 (2002), 15–18.

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

[3] Baker, P., Harman, M., Steinhöfel, K., and Skaliotis, A. Search Based Approaches to Component Selection and Prioritization for the Next Release Problem. In *22nd International Conference on Software Maintenance (ICSM 06)* (Philadelphia, Pennsylvania, USA, September 24-27 2006), pp. 176–185.

[4] Banjotopoulos, C. A Contribution to the Sensitivity Analysis of the Sea-Bed-Structure Interaction Problem for Underwater Pipelines. *Computers & structures 40*, 6 (1991), 1421–1427.

[5] Boehm, B., Abts, C., and Chulani, S. Software Development Cost Estimation Approaches - A Survey. *Annals of Software Engineering 10*, 1 (2000), 177–205.

[6] Boehm, B. W. *Software Engineering Economics*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[7] Christopher Frey, H., and Patil, S. Identification and Review of Sensitivity Analysis Methods. *Risk Analysis 22*, 3 (2002), 553–578.

[8] Deb, K., and Gupta, H. Searching for Robust Pareto-Optimal Solutions in Multi-Objective Optimization. In *Conference on Evolutionary Multi-Criterion Optimization (EMOą́r05)* (2005), vol. 3410, Springer, pp. 150–164.

[9] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on 6*, 2 (2002), 182–197.

[10] Flyvbjerg, B., Holm, M. K. S., and Buhl, S. L. Underestimating Costs in Public Works Projects. *Journal of the American Planning Association 68*, 3 (2002), 279–295.

[11] Flyvbjerg, B., Holm, M. K. S., and Buhl, S. L. How (In)ccurate Are Demand Forecasts in Public Works Projects? *Journal of the American Planning Association 71*, 2 (2005), 131–146.

[12] Gunawan, R., Cao, Y., Petzold, L., and Doyle, F. Sensitivity Analysis of Discrete Stochastic Systems. *Biophysical Journal 88*, 4 (2005), 2530–2540.

[13] Levine, R., and Renelt, D. A Sensitivity Analysis of Cross-Country Growth Regressions. *American Economic Review 82*, 4 (September 1992), 942–63.

[14] Osyczka, A. Multicriteria Optimization for Engineering Design. In *Design Optimization* (1985), pp. 193–227.

[15] Saltelli, A., Chan, K., Scott, E., et al. *Sensitivity Analysis*. Wiley Chichester, 2000.

[16] van den Akker, M., Brinkkemper, S., Diepen, G., and Versendaal, J. Software Product Release Planning Through Optimization and what-if Analysis. *Information and Software Technology 50*, 1–2 (2008), 101–111.

[17] van Veldhuizen, D. A., and Lamont, G. B. Multiobjective Evolutionary Algorithm Test Suites. In *SAC '99: Proceedings of the 1999 ACM Symposium on Applied Computing* (New York, NY, USA, 1999), ACM, pp. 351–357.

[18] Zhang, Y., Harman, M., and Mansouri, A. The Multi-Objective Next Release Problem. In *ACM Genetic and Evolutionary Computation Conference (GECCO 2007).* (London, England, 7 - 11 July 2007).