Today/Future Importance Analysis

Yuanyuan Zhang
CREST Centre
King's College London
London, UK
yuanyuan.zhang@kcl.ac.uk

Enrique Alba
Computer Science
University of Malaga
Malaga, Spain
eat@lcc.uma.es

Juan J. Durillo
Computer Science
University of Malaga
Malaga, Spain
durillo@lcc.uma.es

Sigrid Eldh
Ericsson AB
Kista, Sweden
sigrid.eldh@ericsson.com

Mark Harman
CREST Centre
King's College London
London, UK
mark.harman@kcl.ac.uk

ABSTRACT

SBSE techniques have been widely applied to requirements selection and prioritization problems in order to ascertain a suitable set of requirements for the next release of a system. Unfortunately, it has been widely observed that requirements tend to be changed as the development process proceeds and what is suitable for today, may not serve well into the future. Though SBSE has been widely applied to requirements analysis, there has been no previous work that seeks to balance the requirements needs of today with those of the future. This paper addresses this problem. It introduces a multi-objective formulation of the problem which is implemented using multi-objective Pareto optimal evolutionary algorithms. The paper presents the results of experiments on both synthetic and real world data.

Categories and Subject Descriptors

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

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Pareto optimality, Today/Future, multi-objective genetic algorithms

1. INTRODUCTION

The elicitation and analysis of requirements is an important activity that typically occurs in the early stages of the software engineering development process [22, 24]. The activities associated with requirements have a significant bearing on the whole Software Engineering process and it is widely believed [4] that mistakes and misunderstandings that occur during requirements analysis, have a profound (and consequently expensive) impact upon the entire development process.

Although requirements elicitation is usually one of the first activities to be undertaken during the software development process, it has also been widely observed [11, 14] that requirements are the subject of many changes and revisions as the process of development progresses. This tendency for requirements to change throughout the development process has been regarded as one of the reasons why software development is so difficult and expensive.

Search Based Software Engineering and related optimization techniques have been applied to requirements analysis in order to ascertain a suitable set of initial requirements [32]. This work is often referred to as the Next Release Problem (NRP) [1], the Multi-Objective Next Release Problem (MONRP) [33] or as the process of release planning [15], because it considers the set of requirements to be planned for in the next release of the software system. Release planning is the problem of determining a set of requirements that balance competing constraints, such as cost/value trade-offs [20] and the balance between higher and lower level concerns. A brief overview of work on SBSE for requirements is provided by Zhang et al. [32].

Previous work on the application of SBSE to requirements has focussed on this release planning process and therefore it is concerned largely with the initial stage of requirements optimization; the problem of determining a suitable set or priority ordering for the initial set of requirements for the next release of the system. To the authors' knowledge and according to recent surveys of SBSE [17], there has been no previous work on the application of SBSE to the problem of managing requirements change within the optimization formulation.

This paper seeks to address this problem. We extend previous work on release planning by considering the problem of finding a suitable set of requirements that balances the 'needs of today' (the initial set of requirements to be selected) against the needs for the future. We model this as a multi-objective problem, in which the two objectives of satisfying the needs of today and those of the future are to be balanced. In common with other work on multi-objective SBSE [3, 10, 26, 27, 31, 33], we adopt a Pareto optimal ap-

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

proach, which treats the objectives of today and the future as incomparable objectives.

We report on experimentation with this model formulation to a set of requirements data from Ericsson. This data set has not previously been studied and was collected by Ericsson using its own requirements elicitation process.

The data set was obtained from 14 groups of Ericsson software test engineers, each drawn from different parts of the organization. The data elicited by Ericsson concerns each groups' requirements for a testing tool. The teams give the requirements data for today's testing requirements and those anticipated for the future. The optimization problem is to locate solutions that balance these two objectives and the third objective of reducing the overall cost of the solution chosen.

This paper makes two contributions to the problem of SBSE for requirements optimization. It introduces and formulates the problem of balancing requirements for today and the future and it reports on experiments with this formulation on various data sets including a real-world example from Ericsson. The widely used NSGA-II Pareto optimal multi-objective evolutionary algorithm was used to find Pareto fronts that balance three objectives: *cost*, *value for today* and *value for the future*.

The rest of the paper is organized as follows: Section 2 describes the motivation of this work. In Section 3 the research work is defined formally, while Section 4 describes the data sets used in the empirical studies including one real world data set and 27 combination levels of synthetic data sets. Section 5 defines the fitness function used. Section 6 presents the results of the experiments and discusses the findings. Section 7 describes the context of related work in which the current paper is located. Section 8 concludes.

2. MOTIVATION

The requirements are graded by the respondents as low, medium or high, as suggested in the template for requirement formulation given. However, this degree of importance - the *value* of the requirement might vary over time for a specific stakeholder. For example, one requirement might currently have a low value but it may become very important in future, or *vice versa*. In order to address these scenarios, we introduce *Today/Future Importance Analysis* (T/FIA). In T/FIA, the search technique is adapted to find robust solutions, both for today and for the future. The approach seeks to find a balance between what might be termed 'immediate requirement need' and 'future requirement stability'.

There are two types of requirements change: 1. predictable and 2. unpredictable. 1. Unpredictable requirements change is very hard to deal with; it would require one to 'expect the unexpected'. This paper considers the comparatively easier problem of release planning with respect to a set of 'known likely requirements changes'. 2. Predictable requirements change applies when it is known that the requirements for a system will change over time and the changes required can be predicted to some extent.

Though the requirements changes may be known to be likely at the start of the development or selection process, it will be advantageous to seek a balance between those requirements for meeting today's needs and those for the future. For instance, where a system is to be acquired from a known set of possible candidate choices or a configuration of a system is to be chosen that is to be standardized across an entire organization, there will be a set of choices. These choices can be thought of as a set of requirements to be selected.

In selecting the requirements, there is a need to balance the immediate needs of the organization against those of the future. Since there may be many stakeholders with competing and conflicting ideas about the set of requirements to be chosen, simply balancing the needs of each stakeholder with regard to today's needs is, in itself, a non-trivial SBSE problem [1, 15, 19, 33]. However, the decision maker should also attempt to take account of likely known changes in requirements over time, in making the choice of system. This raises the issue of the independent (but also potentially competing) objective of selecting a set of requirements that is not only optimal for today, but also for the future.

3. REPRESENTATION

It is assumed that for an existing software system, there is a set of stakeholders,

$$C = \{c_1, \ldots, c_m\}$$

whose requirements are to be considered in the development of the next release of the software.

Each stakeholder may have a degree of importance for the company that can be reflected by a weight factor. The set of relative weights associated with each stakeholder c_j $(1 \leq j \leq m)$ is denoted by a weight set:

$$Weight = \{w_1, \ldots, w_m\}$$

$w_1 = w_2 = \cdots = w_m$ if all the stakeholders are treated as equal, that is, the stakeholder weight factor can be ignored.

The set of possible software requirements is denoted by:

$$\Re = \{r_1, \ldots, r_n\}$$

The granularity of requirements can be decomposed hierarchically. At the top level (goals) are a statement of stakeholder needs; the lower levels (refined subgoals) include stakeholder requirements, system requirements, system component requirements and even subsystem component requirements. The lower the level, the more detail is required in order to express the requirement.

The resources needed to implement a particular requirement can be transformed into cost terms and considered to be the associated cost to fulfil the requirement. The resultant cost vector for the set of requirements r_i $(1 \leq i \leq n)$ is denoted by:

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

Usually, different stakeholders have different needs and perspectives. It is assumed that not all requirements are equally important for a given stakeholder. The level of satisfaction for a given stakeholder depends on the requirements that are satisfied in the next release of the software, which provides a *value* to the stakeholders' organizations.

Each stakeholder $c_j (1 \leq j \leq m)$ assigns two types of *value* to each requirement $r_i (1 \leq i \leq n)$ denoted by *value for today*: $v_{today}(r_i, c_j)$ and *value for the future*: $v_{future}(r_i, c_j)$, where $v(r_i, c_j) \in \{low, medium, high\}$ if stakeholder c_j desires implementation of the requirement r_i and $v(r_i, c_j) = ignore$ otherwise.

Accordingly, the *score* of a given requirement r_i can be represented as:

$$score\ for\ today: \quad score_{(i,today)} = \sum_{j=1}^{m} w_j \cdot v_{today}(r_i, c_j)$$

$$score\ for\ the\ future: \quad score_{(i,future)} = \sum_{j=1}^{m} w_j \cdot v_{future}(r_i, c_j)$$

where stakeholder *weight* $w_1 = w_2 = \cdots = w_m$, that is, all the stakeholders are treated as equals in this study. We would like to provide a comprehensive representation of search-based requirements optimization, so the *weight* parameter is not excluded from the model. The '*score*' of a given requirement is represented as its overall '*value*' for the company.

The decision vector $\overrightarrow{x} = \{x_1, \ldots, x_n\} \in \{0, 1\}$ determines the requirements that are to be satisfied in the next release. In this vector, x_i is 1 if requirement i is selected and 0 otherwise. This vector denotes the solution to the problem.

4. DATA SETS

A real world data set and 27 combination random data sets were used to evaluate our approach to T/FIA. The real world data set is taken from Ericsson. It includes questionnaire forms for test management tools, which were completed by 14 stakeholders (each stakeholder was a software testing sub-organization within Ericsson). The test management tool is proposed software to be used for generating, organising and executing the tests (manual or automatic), allowing for requirements tracking, defect tracking and test result reporting.

This questionnaire included 124 requirements for a possible test management tool which was to be selected. The requirements were divided into three major aspects: general, test management and technical requirements. The questionnaire design and collection of data was performed by Ericsson. The details are listed in the Table 1.

To complete the questionnaires, the 14 stakeholders measured how important each requirement is to them in two ways. One is to evaluate the degree of importance for today, the other is the importance for the future. This approach was adopted by Ericsson and not suggested by the author. However, we realized that the information could be useful as a source of analysis for requirements change, and this suggested the T/FIA approach, using a multi-objective approach introduced in this paper.

Each measurement was graded using four levels: *ignore*, *low*, *medium* or *high*. For instance, "The tool shall support project-specific test case parameters, such as test case priority, test environment" was assigned as *medium* for today and *high* for the future by one of the stakeholders, which is a requirement in the "Test Analysis and Design" section of the questionnaire.

In terms of random data sets, "27 combination levels of random data sets" were used. These are the basis of data sets we used in the empirical studies. The "27-random" data set levels were generated using a pseudo random approach, according to distributions of interest. The 'random' data sets therefore allow us to experiment with different data distributions and to compare results with real world data. The synthetic test problems were created by assigning random choices for value and cost. The range of costs were from 1 through to 9 inclusive (zero cost is not permitted). The range of values were from 0 to 5 inclusive (zero value is permitted, indicating that the stakeholder places no value on this requirement).

Table 1: Requirements for Test Management Tools

1.	General Requirement
1.1	Application
1.2	Usability
1.3	System Environment and Installation
1.4	Service and Support
1.5	The Tool Supplier
2.	Test Management Requirement
2.1	Test Planning and Management
2.2	Test Analysis and Design
2.3	Test Artefacts Handling
2.4	TR or Defect Handling
2.5	Requirement Handling
2.6	Test Reporting (actual and trends graphs)
2.7	Configuration Management
2.8	Interfaces
3.	Technical Requirement
3.1	Capacity
3.2	Reliability, Stability and Scalability
3.3	Security

Table 2: 27 Combination Levels of Random Data Sets

	R_{small}	R_{medium}	R_{large}
C_{small}	$\boldsymbol{C_s\ R_s\ D_{low}}$ $C_s\ R_s\ D_m$ $C_s\ R_s\ D_h$	$C_s\ R_m\ D_{low}$ $C_s\ R_m\ D_m$ $C_s\ R_m\ D_h$	$C_s\ R_l\ D_{low}$ $C_s\ R_l\ D_m$ $C_s\ R_l\ D_h$
C_{medium}	$C_m\ R_s\ D_{low}$ $C_m\ R_s\ D_m$ $C_m\ R_s\ D_h$	$C_m\ R_m\ D_{low}$ $\boldsymbol{C_m\ R_m\ D_m}$ $C_m\ R_m\ D_h$	$C_m\ R_l\ D_{low}$ $C_m\ R_l\ D_m$ $C_m\ R_l\ D_h$
C_{large}	$C_l\ R_s\ D_{low}$ $C_l\ R_s\ D_m$ $C_l\ R_s\ D_h$	$C_l\ R_m\ D_{low}$ $C_l\ R_m\ D_m$ $C_l\ R_m\ D_h$	$C_l\ R_l\ D_{low}$ $C_l\ R_l\ D_m$ $\boldsymbol{C_l\ R_l\ D_h}$

For each level, such as $C_s\ R_m\ D_h$, there are three factors involved: the number of requirements, the number of stakeholders and the density of the stakeholder-requirement matrix. This simulates the situation where a stakeholder ranks the choice of requirements (for value) and the cost is estimated to fall in a range, very low, low, medium, high, very high. The number of stakeholders and the number of requirements are divided into three situations, namely, small scale, medium scale and large scale; the density of the stakeholder-requirement matrix is defined as low level, medium and high level. Table 2 lists the combination of all cases schematically. As can be seen in Table 3, the data set divides the range of a variable into a finite number of non-overlapping intervals of unequal width.

Moreover, three distributions of the relationships between

Table 3: Scale Range of '27-random' Data Set Levels

	Small	Medium	Large
No. of Stakeholders	2-5	6-20	21-50
No. of Requirements	1-100	101-250	251-600
	Low	Medium	High
Density of Matrix	0.01-0.33	0.34-0.66	0.67-1.00

Value for today and *Value for the future* for each level were generated, that is (1) *Value* is generated in such a way that it tends to conflict with *today* and *the future*; (2) *Value* is generated in such a way that it tends to agree with *today* and *the future*; (3) *Value* is generated completely randomly.

Any randomly generated, isolated data set clearly cannot reflect real-life scenarios. However, we do not seek to use our pseudo random generation of synthetic data as a substitute for real world data. Rather, we seek to generate synthetic data in order to explore the behavior of our algorithms in certain well defined scenarios. The use of synthetic data allows us to do this within a laboratory controlled environment. Specifically, we are interested in exploring the way the search responds when the data exhibits a presence or absence of correlation in the data.

As well as helping us to better understand the performance and behavior of our approach in a controlled manner, this also allows us to shed light on the real world data, comparing results with the synthetic data. In the literature, Garousi [13] also changed different empirical variation criteria to generate synthetic test models in order to "test the repeatability and scalability aspects of the GA" in the stress testing.

5. FITNESS FUNCTION

The purpose of T/FIA is to provide robust solutions not only in the context of present conditions but also in response to those future changes that can be anticipated. Therefore, three objectives are taken into consideration in order to maximize stakeholders' satisfaction for today, for the future and to minimize the cost of implementation.

The following two fitness functions are considered for maximizing total value for today and future respectively:

$$\text{Maximize } f_1(\overrightarrow{x}) = \sum_{i=1}^{n} score_{i,today} \cdot x_i$$

$$\text{Maximize } f_2(\overrightarrow{x}) = \sum_{i=1}^{n} score_{i,future} \cdot x_i$$

The problem is to select a subset of the stakeholders' requirements which results in the maximum value for the company.

The third fitness function is defined as follows to minimize total cost required for the satisfaction of stakeholder requirements.

$$\text{Minimize } f_3(\overrightarrow{x}) = \sum_{i=1}^{n} cost_i \cdot x_i$$

Since we cannot say whether it is better to be a good solution for today (at the expense of the future) or *vice versa*, it seems natural to treat the objectives as incomparable and to use a Pareto-optimal approach, as we have advocated elsewhere in this paper.

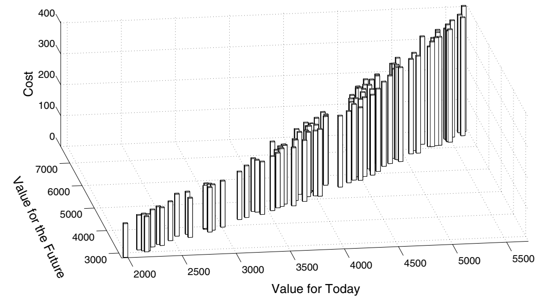


Figure 1: Results from Ericsson Data Set: 124 Requirements, 14 Stakeholders

6. RESULTS AND ANALYSIS

All the experiments were performed using the MATLAB system (*R2007a*). The main programming language is Matlab script and all the data sets were formatted into *.mat* files. The system was installed on an Intel Core 2 Duo processor 2.26 GHz with 4Gb RAM.

The NSGA-II algorithm was chosen for this empirical study. It is a very widely studied 'standard' approach to multi-objective evolutionary optimization. The algorithm was run for a maximum of 1,350,000 fitness function evaluations for each data set. The initial population was set to 300. We used a simple binary GA encoding, with one bit to code for each decision variable (the inclusion or exclusion of a requirement). The length of a chromosome is thus equivalent to the number of requirements. Each experimental execution of each algorithm was terminated when the generation number reached 151 (i.e after 45,000 evaluations). We used tournament selection (the tournament size is 5), single-point crossover and bitwise mutation for binary-coded GAs. The crossover probability was set to $P_c = 0.8$ and mutation probability to $P_m = 1/n$ (where n is the string length for binary-coded GAs). All the figures plotted in the paper show the best results from the 30 independent runs of the algorithm.

The results of the Ericsson data set are plotted in Figure 1. The figure depicts a three-dimensional solution space. The three axes represent the three objectives as *Value for Today*, *Value for the Future* and *Cost*. Each bar denotes an optimal solution on the Pareto front. The location of each bar in the horizontal plane shows the fitness values of first two fitness functions for all the stakeholders. The height of each bar presents the overall cost for each optimal solution.

It can be seen that the overall *Value* for both today and future increase with no bias along the *Cost* axis in the graph. The algorithm produced a Pareto front with a good spread (i.e. high diversity). We provide another view of the results onto the X-Y plane, shown in Figure 2. The cost of solutions is represented as different grey scales. If one slices the graph according to different cost levels, corresponding solutions can be found on the Pareto front. The decision maker can choose the optimal one in different contexts from the alternative Pareto optimal solutions.

In addition, we can analyze the character of data set based on the results. The shape of Pareto front in Figure 1 looks

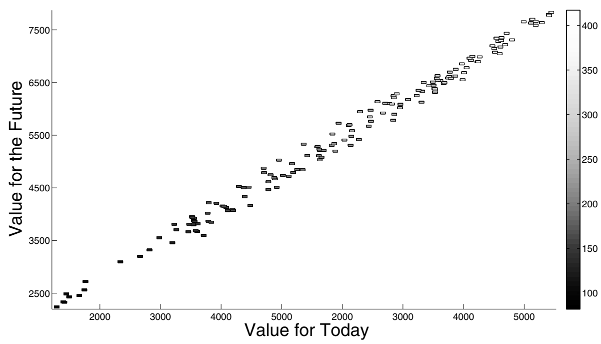


Figure 2: Projection onto the X-Y Plane (Results from Ericsson Data Set)

like a "thin hedge" which bisects the solution space. The same observation can be seen from the projected view on the X-Y plane in Figure 2. The solutions occupy a narrow section of the solution space.

Therefore, from the graph we predict that the requirements' values for today and for the future graded by the stakeholders have a strong correlation. This provides some degree of security that the risk is low compared to a solution in which *value for today* differs greatly from *value for the future*. The Spearman's Rank Correlation [21] statistical analysis test was carried out on the Ericsson data set to assess whether the two *measurement variables* are indeed correlated. The Spearman's Rank Correlation Coefficient, r_s, can take value between -1 and $+1$. $r_s = -1$ means two variables have a perfect negative correlation (as one increases, the other decreases); Accordingly, $r_s = +1$ means two variables have a perfect positive correlation (as one increases, so does the other); $r_s = 0$ means two variables are entirely independent; there is no correlation between them.

In our test $r_s = 0.5697$ which is larger than critical value of r_s at the 95% significance level. This indicates a positive correlation between the *value for today* and *value for the future*. Thus our informed observation from the optimization is also borne out by the statistical analysis. This is a straightforward illustration that the search-based approach not only generates solutions themselves but can also provide the decision maker with useful insights into the data set.

For the results of "27 random data sets", space does not permit us to show all results, nine typical results are illustrated in Figure 3. The results from other data sets were very similar to those shown in the figure. Figure 3 depicts the results from three different scales, small size C_s R_s D_{low}, medium size C_m R_m D_m and large size C_l R_l D_h, in terms of the number of requirements and stakeholders as shown in Table 2 in bold font. Three data sets were included in each scale according to three distributions between *Value for Today* and *Value for the future*.

We also projected the results onto the X-Y plane and the cost of solutions is represented as the different grey scales. In terms of the results of three distributions for *Value*, from each column in the graph, we can see that the "random" distribution made the Pareto fronts widely scattered on the solution space as shown in Figure 3 (a), (d) and (g). The results from the "tending to agreement" distribution, illustrated in Figure 3 (b), (e) and (h), are relatively straight laid, diagonally across the space. From Figure 3 (c), (f) and (i) we can see that the shapes of the Pareto fronts from the "tending to conflict" distribution are in-between the former two.

When compared to the results of three different problem scales, the algorithm produced the Pareto fronts with a good spread for all the scales. The overall *Value* for both today and the future also increase with no bias along the *Cost* axis. Moreover, we can observe that the degree to which the results scatter on the solution space depends on the problem scale. The larger the scale, the wider the range of the solutions scattered.

Another observation of results is the different execution time on three scales. In the paper we measured all the execution time in the 27 data set levels listed in Table 4. As mentioned before, there are three data sets generated in each level according to three distributions for *Value*. The entry listed in the table is the average execution time for three data sets. The unit of time measured is second.

Table 4: CPU Time of 27 Combination Levels of Random Data Sets

	R_{small}	R_{medium}	R_{large}
C_{small}	76.58	275.14	785.56
	66.84	190.74	675.26
	72.69	220.72	784.27
C_{medium}	78.52	274.23	661.10
	72.09	226.39	667.52
	78.38	234.45	780.86
C_{large}	69.50	311.61	742.31
	85.56	253.76	601.99
	75.93	213.11	510.43

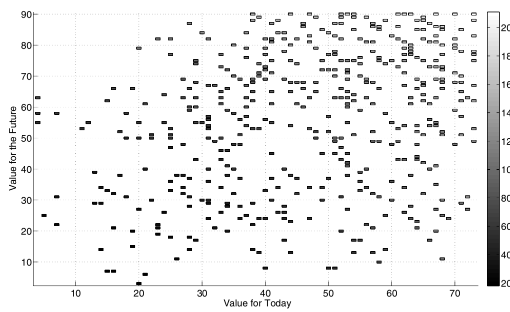
7. RELATED WORK

There has been a recent interest in applying SBSE to requirements problems. A brief overview of recent trends in requirements analysis optimization is provided by Zhang et al. [32].

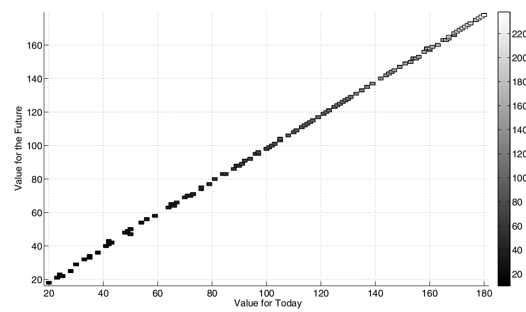
Karlsson et al. [19] were among the first authors to apply optimization techniques to requirements analysis problems. They used the AHP algorithm to optimize requirements orderings, treating the problem as one of prioritizing requirements. This seminal work led to the development of the Focal Point requirements analysis tool, now developed and brought to market by TeleLogic, a subsidiary of IBM.

The problem of requirements optimization can be viewed either as a selection problem or a prioritization problem or both. Bagnall *et al.* [1] formulated requirements analysis as a selection problem, for which they coined the term the 'Next Release Problem (NRP)'. The authors did not give any *value* property to each requirement. They only used an associated *cost*. The task of the work was to find a subset of stakeholders, whose requirements are to be satisfied. The objective was to maximise the cumulative measure of the stakeholder's importance to the company under resource constraints.

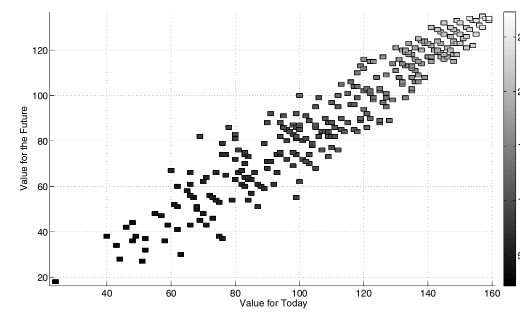
Feather and Menzies [10] applied Simulated Annealing to



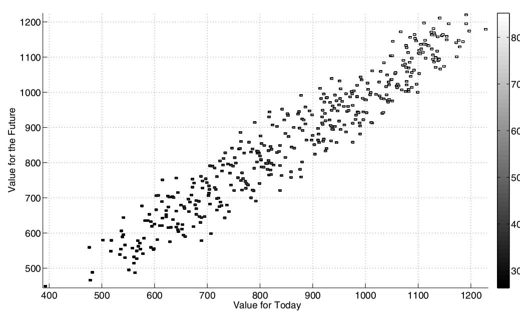
(**a**) Small Scale;
"random" distribution for *Value*



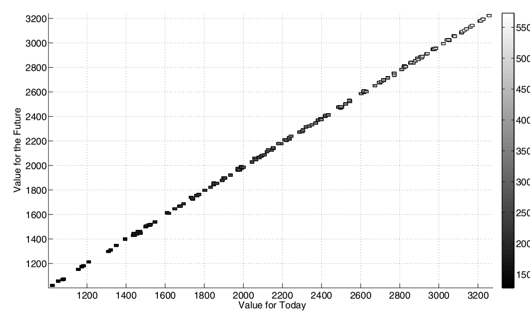
(**b**) Small Scale;
"tending to agreement" distribution for *Value*



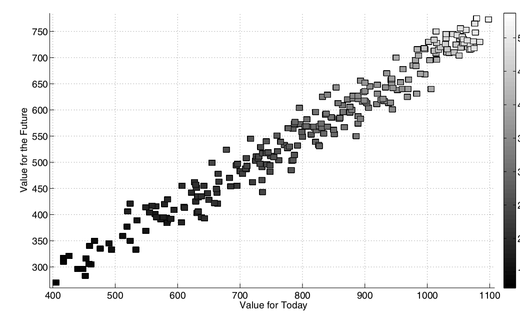
(**c**) Small Scale;
"tending to conflict" distribution for *Value*



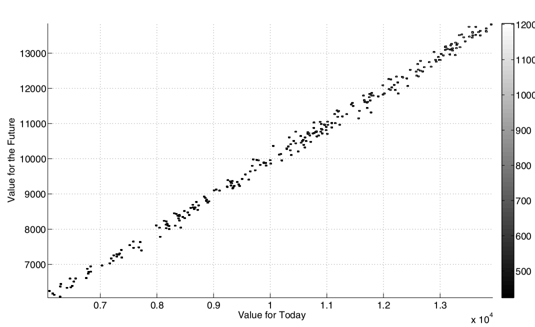
(**d**) Medium Scale;
"random" distribution for *Value*



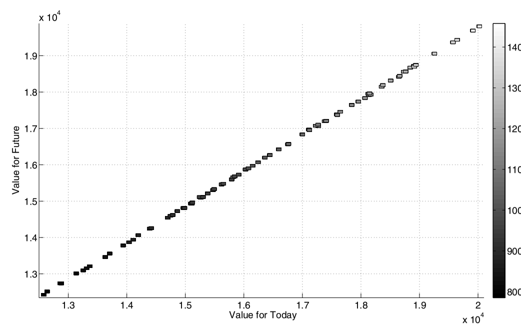
(**e**) Medium Scale;
"tending to agreement" distribution for *Value*



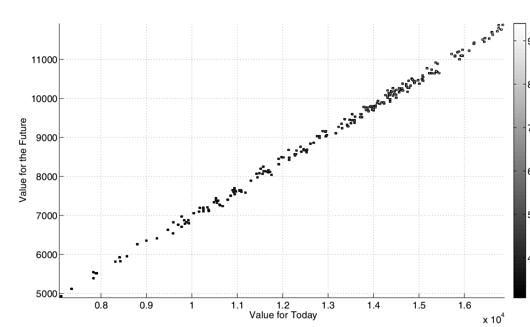
(**f**) Medium Scale;
"tending to conflict" distribution for *Value*



(**g**) Large Scale;
"random" distribution for *Value*



(**h**) Large Scale;
"tending to agreement" distribution for *Value*



(**i**) Large Scale;
"tending to conflict" distribution for *Value*

Figure 3: Results of three scales in 27 combination levels of random data sets

Requirements Analysis problems, introducing an iterative formulation of Requirements Analysis in terms of both selection and optimization. Greer and Ruhe [15] also considered the Requirements Analysis problem as one of selection, introducing the concept of release planning. Ruhe and Saliu [25] presented an Integer Linear Programming (ILP) based method which combined computational intelligence and human negotiation to resolve their conflicting objectives.

Van den Akker et al. [23, 28, 29, 30] further extended the technique and developed an optimization tool based on integer linear programming, integrating the requirements selection and scheduling for the release planning to find the optimal set of requirements with the maximum revenue against budgetary constraints. Harman et al. [17] explored the relationship between SBSE problems in requirements and regression testing in a recent comprehensive survey of work on SBSE.

The problem of optimizing requirements has remained a popular topic areas within SBSE research activity. Baker et al. [2] used Greedy algorithms to address the NRP problem via ranking and selection of candidate software components. Zhang et al. [33] generalized the NRP to the Multi-Objective NRP (MONRP), in order to optimize value and cost. They present the results of an empirical study into the suitability of multi-objective search techniques.

Saliu and Ruhe [26] showed how implementation objectives and requirements objectives could be simultaneously optimized using a multi-objective optimization approach. Finkelstein et al. [12] also used a multi-objective optimization approach to explored requirements assignment fairness, showing how different kinds of 'fairness' can be handled as multiple objectives. Feather et al. [8, 9] explored a set of Pareto visualization techniques that present requirements optimization using Simulated Annealing. Jalali et al. [18] also considered the requirements optimization problem. They optimized a set of requirements for the NASA's Jet Propulsion Laboratory.

In addition, Cortellessa et al. [5, 6, 7] described an optimization framework to provide decision support for COTS and in-house components selection. The Integer Linear Programming (LINGO model solver) based optimization models (CODER, DEER) were proposed to automatically satisfy the requirements while minimising the cost. Harman et al. (GECCO 09) considered the effects of sensitivity of requirements optimization to estimate uncertainty [16].

Like our work in the present paper, authors of previous work have also used Pareto optimal formulations of requirements optimization problems [10, 26, 33]. However, no previous work has addressed the problem of balancing the needs of today's requirements against those that are likely to emerge as requirements change over time in the future. It is from this aspect of the requirements analysis problem that the present paper obtains its primary novelty when compared to the previous work on requirements optimization.

8. CONCLUSIONS

This paper addresses the problem of balancing the requirements of today with those of the future, referred to as the *Today/Future Importance Analysis* (T/FIA). We extend previous work on release planning by considering the problem of finding a suitable set of requirements that balances the 'needs for today' against the 'needs for the future'. A multi-objective formulation of the problem is implemented using multi-objective Pareto optimal evolutionary algorithms. Therefore, the decision maker may have the ability to take account of likely known changes in requirements over time and to make the optimal choices of system.

We report on experimentation with this formulation to a real world data set from Ericsson as well as 27 combination levels of random data sets. The NSGA-II algorithm has good performance, which produces the Pareto front with good spread. According to different cost levels, a number of corresponding solutions can be found on the Pareto front. The software engineer can choose alternative ones in different contexts. Moreover, the T/FIA not only provide optimal solutions themselves, but also yield interesting insight of the data sets. The shape of Pareto fronts reflect the correlation between *value for today* and *value for the future*, which is also supported by the statistical analysis carried out.

9. REFERENCES

[1] A. J. Bagnall, V. J. Rayward-Smith, and I. M. Whittley. The Next Release Problem. *Information and Software Technology*, 43(14):883–890, December 2001.

[2] P. Baker, M. Harman, K. Steinhöfel, 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, Philadelphia, Pennsylvania, 24-27 September 2006. IEEE Computer Society.

[3] M. Bowman, L. C. Briand, and Y. Labiche. Solving the Class Responsibility Assignment Problem in Object-Oriented Analysis with Multi-Objective Genetic Algorithms. Technical Report SCE-07-02, Carleton University, August 2008.

[4] B. H. C. Cheng and J. M. Atlee. Research Directions in Requirements Engineering. In *Proceedings of the 29th International Conference on Software Engineering and Future of Software Engineering (ICSE/FOSE '07)*, pages 285–303, Minneapolis, MN, 19-27 May 2007.

[5] V. Cortellessa, I. Crnkovic, F. Marinelli, and P. Potena. Experimenting the Automated Selection of COTS Components Based on Cost and System Requirements. *Journal of Universal Computer Science*, 14(8):1228–1255, 2008.

[6] V. Cortellessa, F. Marinelli, and P. Potena. Automated Selection of Software Components Based on Cost/Reliability Tradeoff. In *Proceedings of the 3rd European Workshop on Software Architecture (EWSA '06)*, volume 4344 of *LNCS*, pages 66–81, Nantes, France, 4-5 September 2006. Springer.

[7] V. Cortellessa, F. Marinelli, and P. Potena. An Optimization Framework for "Build-or-Buy" Decisions in Software Architecture. *Computers & Operations Research*, 35(10):3090–3106, October 2008.

[8] M. S. Feather, S. L. Cornford, J. D. Kiper, and T. Menzies. Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation. In *Proceedings of the International Workshop on Requirements Engineering Visualization (REV '06)*, pages 10–10, Minnesota, USA, 11 September 2006. IEEE.

[9] M. S. Feather, J. D. Kiper, and S. Kalafat. Combining

Heuristic Search, Visualization and Data Mining for Exploration of System Design Space. In *Proceedings of the 14th Annual International Symposium on Systems Engineering (INCOSE '04)*, Toulouse, France, 20-24 June 2004.

[10] M. S. Feather and T. Menzies. Converging on the Optimal Attainment of Requirements. In *Proceedings of the 10th IEEE International Conference on Requirements Engineering (RE '02)*, pages 263–270, Essen, Germany, 9-13 September 2002. IEEE.

[11] A. Finkelstein. Advanced Software Engineering Course, 2002.

[12] A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, and Y. Zhang. "Fairness Analysis" in Requirements Assignments. In *Proceedings of the 16th IEEE International Requirements Engineering Conference (RE '08)*, pages 115–124, Barcelona, Catalunya, Spain, 8-12 September 2008. IEEE Computer Society.

[13] V. Garousi. Empirical Analysis of a Genetic Algorithm-based Stress Test Technique. In M. Keijzer, editor, *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*, pages 1743–1750, Atlanta, GA, USA, 12-16 July 2008. ACM.

[14] O. C. Z. Gotel and A. Finkelstein. An Analysis of the Requirements Traceability Problem. In *Proceedings of the 1st International Conference on Requirements Engineering (RE '94)*, pages 94–101, Colorado Springs, Colorado, USA, 18-21 April 1994. IEEE Computer Society.

[15] D. Greer and G. Ruhe. Software Release Planning: An Evolutionary and Iterative Approach. *Information & Software Technology*, 46(4):243–253, March 2004.

[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, Montreal, Canada, 8-12 July 2009. ACM.

[17] M. Harman, S. A. Mansouri, and Y. Zhang. Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications. Technical Report TR-09-03, Department of Computer Science, King's College London, April 2009.

[18] O. Jalali, T. Menzies, and M. Feather. Optimizing Requirements Decisions With KEYS. In *Proceedings of the 4th International Workshop on Predictor Models in Software Engineering (PROMISE '08)*, pages 79–86, Leipzig, Germany, 12-13 May 2008. ACM.

[19] J. Karlsson. *A Systematic Approach for Prioritizing Software Requirements*. PhD thesis, Linköping University, 1998.

[20] J. Karlsson and K. Ryan. A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, 14(5):67–74, 1997.

[21] M. G. Kendall and J. D. Gibbons. *Rank Correlation Methods*. Charles Griffin, 5th edition, 1990.

[22] G. Kotonya and I. Sommerville. *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, 1998.

[23] C. Li, M. van den Akker, S. Brinkkemper, and G. Diepen. Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product. In *Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ '07)*, volume 4542 of *LNCS*, pages 93–108, Trondheim, Norway, 11-12 June 2007. Springer.

[24] R. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 6th edition edition, 1992.

[25] G. Ruhe and M. O. Saliu. The Art and Science of Software Release Planning. *IEEE Software*, 22(6):47–53, November 2005.

[26] M. O. Saliu and G. Ruhe. Bi-Objective Release Planning for Evolving Software Systems. In *Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, pages 105–114, Dubrovnik, Croatia, 3-7 September 2007. ACM.

[27] C. L. Simons and I. C. Parmee. A Cross-Disciplinary Technology Transfer for Search-based Evolutionary Computing: from Engineering Design to Software Engineering Design. *Engineering Optimization*, 39(5):631–648, July 2007.

[28] M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal. Flexible Release Composition using Integer Linear Programming. Technical Report UU-CS-2004-063, Institute of Information and Computing Sciences, Utrecht University, December 2004.

[29] M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal. Flexible Release Planning using Integer Linear Programming. In *Proceedings of the 11th International Workshop on Requirements Engineering for Software Quality (RefsQ '05)*, pages 247–262, Porto, Portugal, 13-14 June 2005. Essener Informatik Beitrage.

[30] M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal. Software Product Release Planning through Optimization and What-If Analysis. *Information and Software Technology*, 50(1-2):101–111, January 2008.

[31] S. Yoo and M. Harman. Pareto Efficient Multi-Objective Test Case Selection. In *Proceedings of the 2007 International Symposium on Software Testing and Analysis (ISSTA '07)*, pages 140–150, London, England, 9-12 July 2007. ACM.

[32] Y. Zhang, A. Finkelstein, and M. Harman. Search Based Requirements Optimisation: Existing Work & Challenges. In *Proceedings of the 14th International Working Conference, Requirements Engineering: Foundation for Software Quality (RefsQ '08)*, volume 5025 of *LNCS*, pages 88–94, Montpellier, France, 16-17 June 2008. Springer.

[33] 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, London, UK, 7-11 July 2007. ACM.