Abstract

Context: *In requirements engineering, there will be many different stake holders. Often the requirements engineer has to find a set of requirements that reflect the needs of several different stake holders, while remaining within budget.*

Objective: *This paper introduces an optimisation-based approach to the automated analysis of requirements assignments when multiple stake holders are to be satisfied by a single choice of requirements.*

Method: *The paper reports on experiments using two different multi-objective evolutionary optimisation algorithms with real world data sets as well as synthetic data sets. This empirical validation includes a statistical analysis of the performance of the two algorithms.*

Results: *The results reveal that the Two-Archive algorithm outperformed the others in convergence as the scale of problems increase. The paper also shows how both traditional and animated Kiviat diagrams can be used to visualise the tensions between the stake holders' competing requirements in the presence of increasing budgetary pressure.*

Conclusion: *This paper presented the concept of internal tensioning among multi-stakeholder in requirements analysis and optimisation for the first time. This analysis may be useful in internal negotiations over budgetary allowance for the project.*

Comparing the Performance of Metaheuristics for the Analysis of Multi-Stakeholder Tradeoffs in Requirements Optimisation

Yuanyuan Zhang[a,1,*], Mark Harman[a], Anthony Finkelstein[a], S. Afshin Mansouri[b]

[a] *University College London, Malet Place, London, WC1E 6BT, UK*
[b] *Brelunel University, Uxbridge, Middlesex, UB8 3PH, UK*

Keywords: Pareto optimality, multi-objective genetic algorithms, requirements engineering

1. Introduction

There has been much recent work on search based optimisation for requirements engineering problems [1, 2, 3, 4, 5, 6, 7, 8]. This work has traditionally considered objectives such as cost and value of requirements and sought to maximise the value of the requirements delivered, while minimising the cost [2, 5, 7]. Other candidates for optimisation objectives include fairness [9] and the balance between requirements and implementation constraints [6]. However, no previous work has considered the satisfaction of each stake holder as a separate and competing optimisation objective.

Often, in requirements engineering, there will be many different stake holders [10]. This multi-stakeholder scenario is increasingly important. Often the requirements engineer has to find a set of requirements that reflect the needs of several different stake holders, while remaining within budget. In many development organisations, there may be many stake holders with their own varied, even conflicting views on the sets of requirements to be prioritised.

The paper takes, as its stating point, the question: ‘Which requirements should appear in the next release of the system?’. This question has come to be known as the Next Release Problem (NRP) [1, 3], while the activity of solving this NRP is known as Release Planning [11, 12, 13, 14, 15, 16, 17].

Finding good solutions to the NRP is inherently a complex optimisation problem; the underlying problem being NP-hard [18]. To overcome the difficulties associated with such NP hard Software Engineering problems, Search Based Software Engineering (SBSE) uses metaheuristics to solve complex and highly constrained problems [19]. This approach allows us to search the space of potential requirement allocations (which grows exponentially with the number of requirements). Recent results [20] have demonstrated that these search–based formulations of the NRP can yield human-competitive results, indicating that the optimisation approach holds out the promise of providing automated decision support to the requirements engineer.

This paper introduces a new search based approach to the problem of balancing requirements where there are several stake holders, each with their own view of the importance of different requirement sets.

[*] Corresponding author

Email addresses: `yuanyuan.zhang@cs.ucl.ac.uk` (Yuanyuan Zhang), `m.harman@cs.ucl.ac.uk` (Mark Harman), `a.finkelstein@cs.ucl.ac.uk` (Anthony Finkelstein), `afshin.mansouri@brunel.ac.uk` (S. Afshin Mansouri)

[1] Tel. +44 (0) 20 7679 1056, Fax. +44 (0) 20 3108 5040

The proposed approach uses search-based optimisation techniques to automate the search for optimal or near optimal allocations of requirements that balance these competing stake holder objectives. The search is guided by a fitness function that seeks to satisfy each stake holder using the Pareto optimality.

Sometimes, satisfying one stake holder can only be achieved at the expense of failing to satisfy another. We call this the 'tensioning' between these stake holders and visualise it using an animated Kiviat diagram [21, 22]. In this diagram, visually, we may consider one stake holder 'pulling' the closed polygon solution towards their choice of requirements, creating this 'tension'.

The technical novelty of the approach lies in the way in which stake holders are each mapped to a separate optimisation objective. No previous work has treated the satisfaction of each stake holder as an optimisation objective in its own right. We argue that this 'one-objective-per-stakeholder' formulation is a natural one for an optimisation-based approach. It allows us to make the tensioning between the stake holders explicit and to explore the way in which this tension responds to different levels of budget.

It is important that the presentation of the results of optimisation are presented to the decision maker in a form that is intuitive and free from technical detail. Many decision makers are not, themselves, technical experts. Rather, they often are managers and business analysts, who may rely upon automated decision making support, but cannot be expected to familiarise themselves with algorithmic details. Though we use sophisticated multi objective evolutionary algorithms to search for near optimal solutions among the enormous sets of potential solutions, our approach requires no familiarity with these techniques on the part of the decision maker. The paper's use of animated Kiviat diagrams for this purpose is also novel. This animated visualisation is used to render the results in an intuitive manner to the decision maker.

The primary contributions of the paper are thus summarised as follows:

1. The paper formulates the Next Release Problem as a Multi-Objective Optimisation Problem (MOOP), treating each stake holders' set of requirements as a separate objective to calculate and investigate the extent of satisfaction for each stake holder. This is the first paper to propose this one-objective-per-stakeholder formulation.
2. The proposed approach caters for tensioning of the relationship between the stake holders and the resources as well as the natural internal tensioning among the stake holders. These scenarios let the decision maker explore and optimise the tradeoffs allowed by the instantiation of the problem with which they are faced.
3. The paper presents an empirical study and statistical analysis in which metaheuristic search techniques are applied to the new formulations of the problem. The results from the empirical study suggest that the Two-Archive algorithm outperforms the NSGA-II algorithm.
4. The approach also provides a simple method to visualise the discovered solutions in the multi-dimensional solution space. Though these visualisations are not novel in themselves, the paper is the first to use them for visualisation of requirement optimisation spaces.

The rest of the paper is organised as follows: In Section 2 the research problem is defined formally. Section 3 introduces the search algorithms applied and algorithmic tuning. Section 4 describes the data sets used, environment and performance metrics for the search algorithms. Sections 5 presents the results of the experiments and discusses the findings from these experiments. Section 6 describes related work giving a context for the current paper and Section 7 considers threats to the validity of the findings in the empirical study. Section 8 concludes the paper.

2. Problem Formulation

This section gives definitions and characteristics of the MONRP problem as an extension of the traditional NRP model.

2.1. NRP Model

It is assumed that for an existing software system, there is a set of stake holders, $C = \{c_1, \ldots, c_m\}$ whose requirements are to be considered in the development of the next release of the software.

In this paper we remain purposefully agnostic about the identity and make up of a 'stake holder'. In the Motorola case study, a stakeholder corresponds to an entire organisation; one with which Motorola as a relationship. However, a stakeholder might also, equally, be an individual user in some other scenario. It is even possible for two or more stakeholders to reside within a single person reflecting two or more different roles that the person fulfils within an organisation. Our approach applies equally well in each of these different usage scenarios.

At one extreme, where a stakeholder is an entire organisation, some aggregation of requirement requests must be undertaken as a preliminary step in order to reach a corporate view on requirement desirability. At the other extreme, where all stake holders are merely the different perspectives that a single individual brings to their project through their different roles or use cases, our approach allows such a 'schizophrenic stakeholder' to explore the internal tensions between their possibly conflicting desires.

The set of possible software requirements is denoted by: $\Re = \{r_1, \ldots, r_n\}$. In order to satisfy each requirement, some resources need to be allocated. The resources needed to implement a particular requirement can be transformed into cost terms and considered to be the associated cost to fulfill the requirement. Typically, these cost values are estimated, as is the case with the real world case studies presented later on. The resultant cost vector for the set of requirements $r_i (1 \leq i \leq n)$ is denoted by: $Cost = \{cost_1, \ldots, cost_n\}$.

It is assumed that all requirements are not equally important to a given stake holder. The level of satisfaction for a given stake holder depends on the requirements that are satisfied in the next release of the software, which provide $value$ to the stake holders' organisations. Each stake holder $c_j (1 \leq j \leq m)$ assigns a *value* to requirement $r_i (1 \leq i \leq n)$ denoted by: $value(r_i, c_j)$ where $value(r_i, c_j) > 0$ if stake holder j desires implementation of the requirement i and 0 otherwise.

$$Value = \left\{ \begin{array}{cccccc} v_{1,1} & v_{1,2} & \cdots & v_{1,i} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,i} & \cdots & v_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ v_{j,1} & v_{j,2} & \cdots & v_{j,i} & \cdots & v_{j,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{m,i} & \cdots & v_{m,n} \end{array} \right\}$$

Thus, each stake holder c_j has a subset of requirements that they expect to be satisfied, denoted by R_j

$$\text{such that} \quad R_j \subseteq \Re, \quad \forall\, r \in R_j \quad value(r, c_j) > 0$$

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. n in the decision vector $\overrightarrow{x}$ is the number of all requirements. In this vector, x_i is

1 if requirement i is selected and 0 otherwise. This vector denotes the solution to the problem. These definitions are based on those used previously in our work on NRP [23].

It should be noted that the ultimate decision as to which requirements should be included in the next release of the software resides with the requirements engineer; such decisions are naturally a mater for human judgement in the final analysis. However, the algorithms used in this and related work on search based requirements optimization can provide insights into the choices available and thereby inform and explore the space of near optimal decision choices from among the inconceivably large number of possibilities. The algorithms used in search based requirements optimization are thus intended to provide decision support, but they can never replace the decision maker.

2.2. Multi-stakeholder Objective Formulation

In the new multi-objective formulation introduced in this paper, each stake holder is taken into consideration as a separate objective in order to maximise each stake holder's satisfaction. That is, the number of objectives in the problem is the same as the number of stake holders. We treat *cost* as a budgetary constraint. This allows us to:

1. Explore the relationship between stake holders and resources; we can understand the extent to which requirement assignments meet stake holder expectations under varying budget allocation conditions;
2. Analyse the internal tensions between the stake holders; which stake holder can easily be satisfied? whose requirements are hard to fulfil? when does pleasing one stake holder cause us to have to displease another?

Evolutionary multi-criteria optimisation has traditionally concentrated on problems comprised of two or three objectives. Our formulation comprises of a relatively large number of objectives. Such problems pose new challenges for algorithm design, visualisation and implementation. In multi-objective evolutionary search, the populations are likely to be largely composed of non-dominated solutions.

The following objective function is used for maximising the total value to each stake holder, expressed as a percentage. For each stake holder c_j, we seek to maximise their expected satisfaction subject to the available budget. More formally,

$$\text{Maximise} \qquad \frac{\sum_{i=1}^{n} value(r_i, c_j) \cdot x_i}{\sum_{r \in R_j} value(r, c_j)}$$

$$\text{subject to} \qquad \sum_{i=1}^{n} cost_i \leq B, \qquad B > 0$$

In the rest of the paper we refer to this term to be maximised as 'stake holder's satisfaction'.

The problem is to select a subset of the stake holders' requirements which results in the maximum percentage of the requirements' *value* that each stake holder expected, while falling within the budgetary constraint. This problem is a multi-objective *knapsack* problem, which is known to be NP-hard. The fitness value of the non-dominated solutions on the Pareto front can be visualised by the Kiviat diagrams [21, 22]. Examples will be presented in detail later on.

3. The Algorithms

This section describes the search algorithms used in our approach. The paper presents results from two search techniques namely: the NSGA-II algorithm and the Two-Archive algorithm. These techniques are also compared with the Random Search to verify viability of applying computationally expensive search techniques for the Multi-stakeholder Objective Formulation.

3.1. The Pareto-Optimal Front

The multi-objective search algorithms used in the paper are based on the concept of dominance to solve MOOPs. Each solution is defined as a vector of decision variables $\overrightarrow{x}$. It is assumed that there are M objective functions $f_i(\overrightarrow{x})$ where $i = 1, 2, \ldots, M$. The objective functions are a mathematical description of performance criteria. The aim to find a set of values for the decision variables that optimises a set of objective functions.

A decision vector $\overrightarrow{x}$ is said to dominate a decision vector $\overrightarrow{y}$ (also written as $\overrightarrow{x} \succ \overrightarrow{y}$) iff:

$$f_i(\overrightarrow{x}) \geq f_i(\overrightarrow{y}) \ \forall\, i \in \{1, 2, \ldots, M\};$$

and

$$\exists\, i \in \{1, 2, \ldots, M\} \mid f_i(\overrightarrow{x}) > f_i(\overrightarrow{y}).$$

All decision vectors that are not dominated by any other decision vector are called *non-dominated* or Pareto-Optimal and constitute the Pareto-Optimal Front. These are solutions for which no objective can be improved without detracting from at least one other objective.

Using Pareto optimality, it is not possible to measure ‘how much’ better one solution is than another, merely to determine whether one solution is better than another. When searching for solutions to a problem using Pareto optimality, the search yields a set of solutions that are non-dominated. That is, each of the non-dominated set is no worse than any of the others in the set, but also cannot be said to be better. Any set of non-dominated solutions forms a Pareto front.

Consider Figure 1, which depicts the computation of Pareto optimality for two imaginary fitness functions (Objective 1 and Objective 2). The longer the search algorithm is run, the better the approximation becomes to the real Pareto front. In the figure, points $S1$, $S2$ and $S3$ lie on the Pareto front, while $S4$ and $S5$ are dominated.

3.2. NSGA-II

The Non-dominated Sorting Genetic Algorithm-II (NSGA-II), introduced by Deb et al. [24] is an extension to an earlier Multi-Objective EA called NSGA developed by Srinivas and Deb [25]. NSGA-II incorporates elitism to maintain the solutions of the best front found. The rank of each individual is based on the level of non-domination. The NSGA-II is a computationally efficient algorithm whose complexity is $O(mN^2)$, compared to NSGA with the complexity $O(mN^3)$, where *m* is the number of objectives and *N* is the population size.

The population is sorted using the non-domination relation into several fronts. The process is described in Algorithm 1. Each solution is assigned a fitness value according to its non-domination level. In this way, the solutions in better fronts are given higher fitness values. The NSGA-II uses a measure of crowding distance to provide an estimation of the density of solutions belonging to the same front,

Algorithm 1: NSGA-II (fast-non-dominated-sort) [26] Deb (2001)

```
input  : $R_t$
output: $F$
for each individual(p)$\in R_t$ do
    Set $S_p = \varnothing$ // the set of individuals dominated by individual $p$
    Set $n_p = 0$ // the number of individuals that dominate individual $p$
    for each individual(q)$\in R_t$ do
        if individual(p) dominates individual(q) then
            Let $S_p = S_p \cup \{\text{individual}(q)\}$
        end
        else if individual(q) dominates individual(p) then
            Let $n_p = n_p + 1$
        end
    end
    if $n_p == 0$ then
        Set $P_{rank} = 1$
        Let $F_1 = F_1 \cup \{\text{individual}(p)\}$
    end
end
Set $i = 1$
while $F_i \neq \varnothing$ do
    Set $Q = \varnothing$
    for each individual(p)$\in F_i$ do
        for each individual(q)$\in S_p$ do
            Let $n_q = n_q - 1$
            if $n_q == 0$ then
                Let $q_{rank} = i + 1$
                Let $Q = Q \cup \{\text{individual } q\}$
            end
        end
    end
    Let $i = i + 1$
    Let $F_i = Q$
end
```

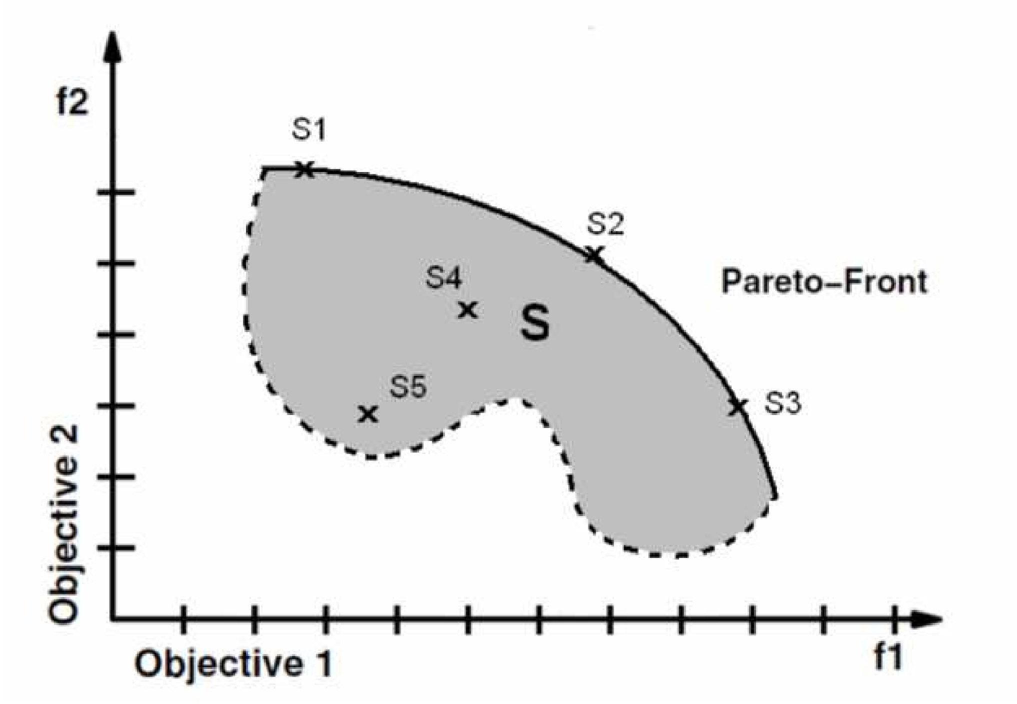


Figure 1. Pareto-Optimal Front

which is described in Algorithm 2. This parameter is used to promote diversity within the population. Solutions with higher crowding distance are assigned a higher fitness compared to those with lower crowding distance, thereby avoiding the use of 'fitness sharing factor' with its associated computational cost [27].

The algorithm progresses as follows. Initially, a random population P_0 with size N is created. Tournament selection, single-point crossover, and bitflip mutation operators are used to create a child population Q_0 of size N [26]. The main loop is described in Algorithm 3.

3.3. Two-Archive

The Two-Archive algorithm was introduced by Praditwong and Yao [28]. It substitutes the new dominated solutions for the existing dominated solutions introduced by PESA [29]. Truncation is performed at the end of archiving non-dominated individuals, as it is in NSGA-II [24] and SPEA2 [30].

In the algorithm there are two archives used to collect and keep the candidate solutions in the population: the convergence archive (CA) and diversity archive (DA). If a non-dominated solution selected from the population dominates some other member in either archive, it enters the CA and the dominated members are deleted, otherwise enters the DA without deleting any members in the archives.

The total combined size of CA and DA is fixed but the size of each archive may vary. When the number of members in the archives exceeds the capacity of archives, the member of DA with the shortest distance to some member in CA is deleted until the total size falls below a predefined threshold size.

The details of the Two-Archive algorithm are shown in Algorithm 4 and Algorithm 5.

Algorithm 2: NSGA-II (crowding-distance-assignment) [26]   Deb (2001)

```
input  : $F$
output: $F$
Set $l = |F|$
for each individual(p)$\in F$ do
    Set $D_p = 0$
end
for each objective $m$ do
    Let $F = sort(F, m)$
    Let $D_1 = D_l = \infty$
    for $p = 2$ to $(l - 1)$ do
        Let $D_p = D_p + (D_{p+1}.m - D_{p-1}.m)/(f_m^{max} - f_m^{min})$
    end
    // $D_p.m$ is the value of the $m$th objective function for individual $p$
    // $f_m^{max}$ and $f_m^{max}$ are the maximum and minimum values of the $m$th objective function.
end
```

Algorithm 3: NSGA-II (main loop) [26]   Deb (2001)

```
input  : $t = 0$;  $P_0$ and $Q_0$
output: $F_{t-1}$
while $t <=$MAX_GENERATION do
    Let $R_t = P_t \cup Q_t$
    Let $F =$ fast-non-dominated-sort $(R_t)$ //$F = (F_1, F_2, ..., F_i, ...)$
    Let $P_{t+1} = \varnothing \ \ and \ \ i = 1$
    while $|P_{t+1}| + |F_i| \leqslant N$ do
        Apply crowding-distance-assignment$(F_i)$
        Let $P_{t+1} = P_{t+1} \cup F_i$
        Let $i = i + 1$
    end
    $Sort(F_i, \prec_n)$
    Let $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$
    Let $Q_{t+1} =$ make-new-pop$(P_{t+1})$
    Let $t = t + 1$
end
```

Algorithm 4: The Two-Archive Algorithm (the main loop)   Praditwong and Yao (2006)

```
Initialise the population
Initialise archives to the empty set
Evaluate initial population
Set t=0
repeat
    Collect non-dominated individuals in archives
    Select parents from archives
    Apply genetic operators to generate a new population
    Evaluate the new population
    t=t+1
until $t == MAX\_GENERATION$;
```

3.4. Random Search

We also applied the Random Search technique to the Multi-Stakeholder tradeoff problem. This is merely a 'sanity check'; all metaheuristic algorithms should be capable of comfortably outperforming Random Search for a well-formulated optimisation problem. Thus the Random Search technique was given the same number of fitness evaluations as the other algorithms to provide a useful lower bound benchmark for measuring the other algorithms' performance.

3.5. Algorithmic Tuning

In order to facilitate replication of our experiments, this section lists the parameter tuning options we used. All approaches were run for a maximum of 50,000 fitness function evaluations. The initial population was set to 500. 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 101 (i.e after 50,000 evaluations). All genetic approaches used tournament selection (the tournament size is 5), single-point crossover and bitwise mutation for binary-coded GAs. All the chromosome cut points were generated randomly. 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).

A number of assumptions have been made regarding the operators as well as parameter values of the search algorithms, which can influence the results. However, all assumptions are the same for all algorithms to minimise the influence of environmental factors during experimentation. The performance of the algorithms could have been further improved by individual fine tuning empirically or through systematic experimentation. However, such fine tuning would inhibit a fair comparison of the two algorithms studied.

For the Two-Archive algorithm, the total capacity of the archives was set to 500. The algorithm selects parents from both archives to the mating pool. An archive was chosen with a probability that was set to 0.8.

The Random Search technique was afforded an 'effort budget' that was set to the same number of fitness evaluations as used by the other algorithms in order to provide a lower bound benchmark for

Algorithm 5: Collect Non-Dominated Individuals in archives   Praditwong and Yao (2006)

```
for i=1 to popsize do
    if individual(i) is non-dominated solution then
        if no member in both archives can dominate an individual(i) then
            Set individual(i).Dflag=0
            if individual(i) dominates any member in both archives then
                Set individual(i).Dflag=1
                Delete dominated member
            end
            if member(i).Dflag==1 then
                Add individual(i) to Convergence Archive(CA)
            else
                Add individual(i) to Diversity Archive(DA)
            end
        end
    end
end
// Remove Strategy
if sizeof(CA)+sizeof(DA)¿limit then
    for i=1 to sizeof(DA) do
        DA(i).length=maxreal
        for j=1 to sizeof(CA) do
            if DA(i).length¿Dist(CA(j),DA(i)) then
                DA(i).length=Dist(CA(j),DA(i))
            end
        end
    end
    repeat
        Delete the member of DA with the shortest length
    until sizeof(DA)+sizeof(CA)==limit;
end
```

measuring the other algorithms' performance.

Each algorithm was executed 20 times for each data set to allow for the application of statistical tests for the significance of algorithmic differences. In addition, there are 27 variations in the second data set, so each algorithm was run 20 times for each variation and the data subset was regenerated randomly to produce a unique data set for each run. In total 8.7×10^6 fitness evaluations were computed. Each of these denotes a possible assignment of requirements.

Tournament selection (the tournament size is 5) is used in the NSGA-II algorithm. In the Two-Archive algorithm, parents are selected from both archives (convergence archive and diversity archive) with a probability set to 0.8.

4. Experimental Set Up

This section describes the test data sets used to compare the performance of the NSGA-II algorithm with the Two-Archive and Random Search algorithms.

4.1. Data Sets Used

There are three data sets used in the experiments to perform multi-stakeholder analysis. The first data set was used in [31], which was generated randomly according to the problem model. 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 stake holder places no value on this requirement). This simulates the situation where a stake holder ranks the choice of requirements (for value) and the cost is estimated to fall in a range, very low, low, medium, high, very high. Our practical experience of requirements analysis suggests that stake holders prefer such a coarse grained scale. While a finer level of granularity may be more theoretically interesting, we found that, in practice, stake holders were uncomfortable with fine-grained value assignments.

In the first data set, the values and costs of which are assigned by the rules described. The number of stake holders and the number of requirements are divided into three situations, namely, small scale, medium scale and large scale; the density of stakeholder-requirement matrix is defined as low level, medium and high level. Table 1 lists the combination of all cases schematically. As can be seen in Table 2, the data set divides the range of a variable into a finite number of non-overlapping intervals of unequal width. Therefore the three algorithms were applied to 27 test problem sets. We call this the '27-random' data set to distinguish it from the real world data sets considered later on.

The second data set is provided by Motorola [32]. The Motorola data set concerns a set of 35 requirements for hand held communication devices. The stake holders are four mobile telephony service providers, each of which has a different set of priorities with respect to the features that they believe ought to be included in each handset.

The third data set is taken from Greer and Ruhe [3]. The Greer & Ruhe data set has 5 stake holders and 20 requirements. The Greer & Ruhe data set does not contain the information about the cost of each requirement. For the purpose of feeding this useful industrial data into our algorithm, the cost of the requirements were generated randomly within the range from 10 to 1100, following a Gaussian distribution.

Table 1. 27 combination random data sets [31]

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

Table 2. Scale Range of '27-random' data set [31]

	Small	Medium	Large
No. of stake holders	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

4.2. Environment

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 or .dat files. The system was installed on an Intel Core 2 Duo processor 2.26 GHz with 4Gb RAM. This information is unremarkable. It is provided merely in order to support replication of the experiments.

4.3. Performance Metrics for Search-based Techniques

A number of performance metrics suggested in the literature [33, 26, 34] show that the quality of a set of non-dominated solutions produced can be measured in terms of convergence and diversity. The approximated solutions must obtain good convergence to the *real* Pareto front, and good diversity along the front in population members.

It is usually necessary to know the location of the real Pareto front. In the case of requirements selection and optimisation problems, the real Pareto front is unknown. Therefore, a reference Pareto front was constructed and used to compare the Pareto fronts produced by different algorithms. Consisting of the best solutions of each technique, the reference Pareto front denotes the best available approximation to the *real* Pareto front. The Pareto fronts generated by the different search techniques may partly contribute to the reference Pareto front.

The reference pareto front is the best pareto front obtainable using all of the algorithms in concert. It is formed from the union of the pareto fronts of each algorithm. From this union, any dominated solutions are removed. The front so-formed is a 'reference' in the sense that it contains the best results obtainable by any and all of the algorithms. As such, it is the best approximation available from the experiments against which to compare each algorithm.

In the paper, three performance metrics were applied to measure the quality of the Pareto fronts produced by the search-based optimisation algorithms:

4.3.1. Metrics for Convergence

1. The Euclidean distance *C* between approximated solutions and reference Pareto front measures the extent of convergence to a known set of solutions.

$$C = \frac{\sum_{i=1}^{N} d_i}{N}$$

   where *N* is the number of solutions obtained by an algorithm. d_i is the smallest Euclidean distance of each solution i to the reference Pareto front. The smaller the calculated value of *C*, the better the convergence. This metric $C = 0$ if the obtained solutions are exactly on the reference Pareto front.

2. The percentage *P* of the solutions on the reference Pareto front, namely the solutions that are not dominated by the reference Pareto front, provides a quantitative comparison between algorithms. The larger the number, the better the convergence.

$$P = \frac{num}{NUM}$$

   where *num* is the number of solutions that are obtained by an algorithm and are on the reference Pareto front and *NUM* is the total number of solutions on the reference Pareto front.

4.3.2. Metric for Diversity

The metric Δ [33] measures the extent of distribution in the obtained solutions and spread achieved between approximated solutions and the reference Pareto front.

$$\Delta = \frac{\sum_{k=1}^{M} d_k + \sum_{j=1}^{N-1} |d_j - \overline{d}|}{\sum_{k=1}^{M} d_k + (N-1)\overline{d}}$$

where $k (1 \leq k \leq M)$ is the number of objectives for a multi-objective algorithm. d_k is the Euclidean distance to the extreme solutions of the reference Pareto front in the objective space. *N* denotes the number of solutions obtained by an algorithm. $d_j (1 \leq j \leq N-1)$ is the Euclidean distance between consecutive solutions, $\overline{d}$ is the average of all the distance d_j.

The smaller the value of Δ, the better the diversity. This metric $\Delta = 0$ if all the obtained solutions have perfect distribution ($d_j = \overline{d}$) and also include all the extreme solutions on the reference front ($d_k = 0$).

5. Results and Analysis

This section reports results of the experiments from synthetic and real data sets described in Section 4.

5.1. Results from Random Data Sets

In the '27-random' data set, the number of stake holders, the number of requirements and the density of the stakeholder-requirement matrix are the three determining factors used to generate each subset of random data. Table 2 shows the variation range of each factor. The information contained in the data set from Motorola was used as a reference and the maximum number of requirements was set to 600.

Table 3. Rank Order for Convergence

	winner	runner up	loser
Random Search	0%	0%	100%
Two-Archive	95.19%	4.81%	0%
NSGA-II	7.04%	92.96%	0%

Table 4. Solutions on the Reference front

Random Search	Two-Archive	NSGA-II
2.68%	94.57%	38.25%

The results are shown in Table 3 and Table 4. For the '27-random' data set, 540 (27×20) runs were executed in the empirical study. The distances from the Pareto front generated by each algorithm to the reference Pareto front, namely, the value of convergence were calculated. The smaller the value of the distance, the better the convergence of the algorithm. The three algorithms were ordered by the value of the convergence in each run. The cumulative total number of orderings for each algorithm was recorded. In other words, we sought to understand how many times a specific algorithm was ranked in a specific position in the 540 runs respectively.

Table 3 shows the proportion of each of the three algorithms in the rank positions: winner, runner up and loser. The Two-Archive algorithm was in the first rank in more than 95% of the cases. It performs the best in the three algorithms overall.

At the same time, the number of the solutions on the Pareto front of each algorithm and the number of the solutions on the reference Pareto front were also recorded to calculate the proportion of the non-dominated solutions on the reference front. As shown in the Table 4, a substantial proportion of the solutions on the reference front are from the results of the Two-Archive algorithm.

5.2. Results from Real Data Sets

The aim of the empirical studies is to compare the performance of the algorithms, and to explore the effect of different resource limitations on conflicting objectives. We considered budgets increasing in steps of 5% from 0% to 100%, where a budget of x% means x% of the cost of all requirements are provided. For statistical investigation of the performance of the algorithms we used three budget levels: 30%, 50% and 70%.

Figures 2 and 3 present Kiviat diagrams that address the two research questions proposed in Section 2. From the Kiviat diagrams depicted in the figures, the number of axes indicates the number of the stake holders. For example, in Figure 2, there are four stake holders, so four axes. This is also the number of the objectives in the Multi-Stakeholder Objective Formulation. Each axis also, therefore, represents an objective (the degree to which the corresponding stake holder is satisfied as a percentage). The intersection points of a closed solid polygon to the axis in the Kiviat diagrams denote the fitness values of the non-dominated solutions.

Animated versions of these Kiviat diagrams (see supplementary GIF files) show the evolution of stake holders' satisfaction as the budget allowance increases from 0 to 100%. The animations provide a vivid dynamic view of the way in which stake holder satisfaction improves unevenly for each stake holder as the budget constraints become more relaxed.

In the paper we can only show 'snap shots' of the animations at 3 illustrative choices for budget. Figure 2 (a), (b) and (c) show the results of the data set from Motorola with 30%, 50% and 70% of resource limitations respectively. The Figure 3 (a), (b) and (c) show the results of the Greer and Ruhe data set.

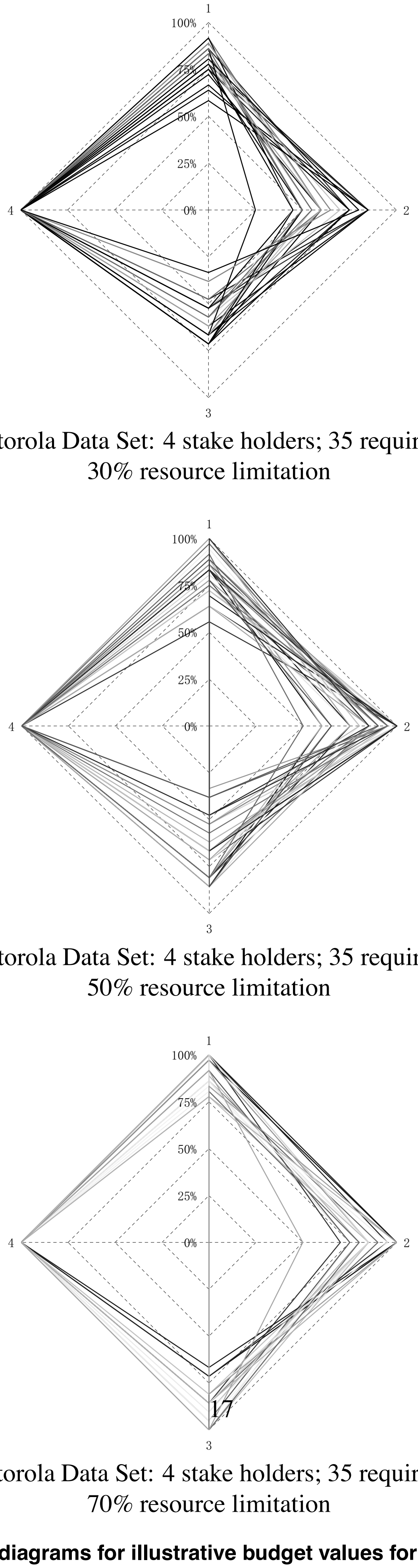
The internal tensioning between the stake holders is shown as some stake holders' demands are easy to satisfy. For example, stake holder 4 in the Motorola data set, whose fitness values can quickly converge to 100% regardless of the satisfaction of the other stake holders. By contrast, stake holder 3 is relatively hard to satisfy, while maintaining the other stake holders' satisfaction. The results from the Greer and Ruhe data set show that there is comparatively little tension between those stake holders.

Figure 4 (a) and (b) provide the plots of the growth in stake holder satisfaction with increasing budgets, which are the results of the data sets from Motorola and Greer and Ruhe generated by the NSAG-II algorithm. The budgetary resource has been shown from 0% to 100% and the step value is 5%. So there are 21 quantified budgetary levels in total. Every stake holder's average satisfaction in 20 runs was calculated in each budgetary resource level.

From the figures we can see, the stake holders' satisfaction began with 0% where there were no available resources and finally all of them converged to 100% when the budgetary resource level reached 100%. There is a remarkable difference in the growth trend of stake holders' satisfaction between Figure 4 (a) and (b). The four stake holders illustrated in Figure 4 (a) have very different satisfaction degrees for each resource level. Such as, the percentages of satisfaction of stake holder 3 (represented by ▲) are the lowest when the budgetary resource levels are between 5% and 55%. The other three stake holders also have greater variation in satisfaction degree compared to the results for the stake holders in the Greer and Ruhe data set in Figure 4 (b). The five stake holders' satisfactions show a relatively consistent and stable growth trend.

The ANOVA statistical analysis technique was used to analyse the diversity and the convergence of the results for the two sets statistically. The null hypothesis is that all three algorithms have the same performance. Rejection of the null hypothesis by ANOVA analysis can, however, only tell us whether the three algorithms' performance were significantly different to one another. To explore further, to see where the differences lie, a multiple–comparison procedure was performed. The LSD (Least Significant Difference) method was employed in a multiple comparison to compare the three algorithms, pairwise.

We used the SPSS statistical application [35] to perform ANOVA analysis. There are two independent variables (called factors) in this case: diversity and convergence, so a two-way ANOVA was run using

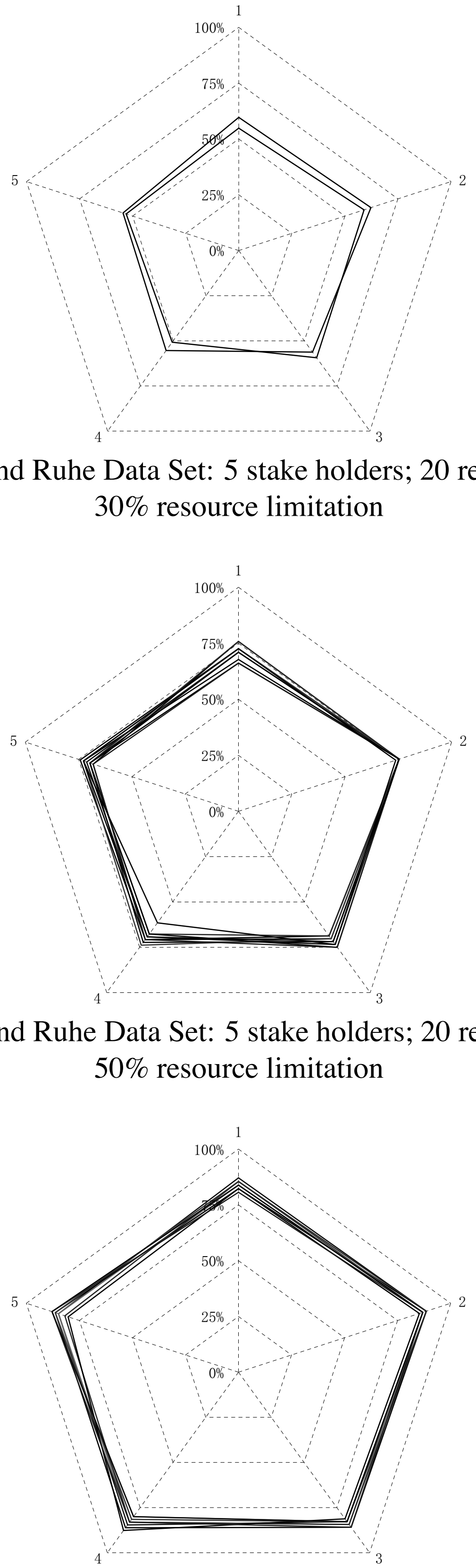


(**a**) Motorola Data Set: 4 stake holders; 35 requirements
30% resource limitation

(**b**) Motorola Data Set: 4 stake holders; 35 requirements
50% resource limitation

(**c**) Motorola Data Set: 4 stake holders; 35 requirements
70% resource limitation

Figure 2. Kiviat diagrams for illustrative budget values for Motorola Data Set

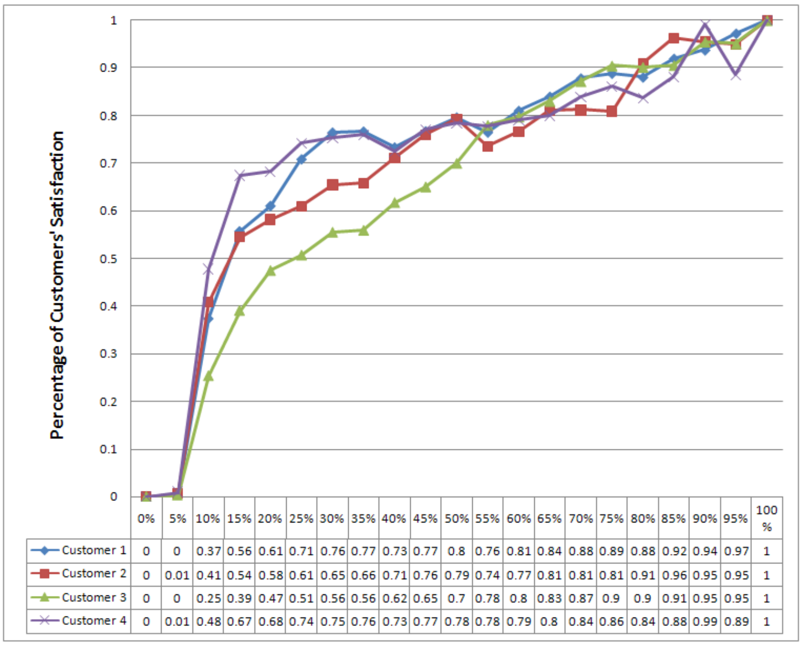


(**a**) Greer and Ruhe Data Set: 5 stake holders; 20 requirements
30% resource limitation

(**b**) Greer and Ruhe Data Set: 5 stake holders; 20 requirements
50% resource limitation

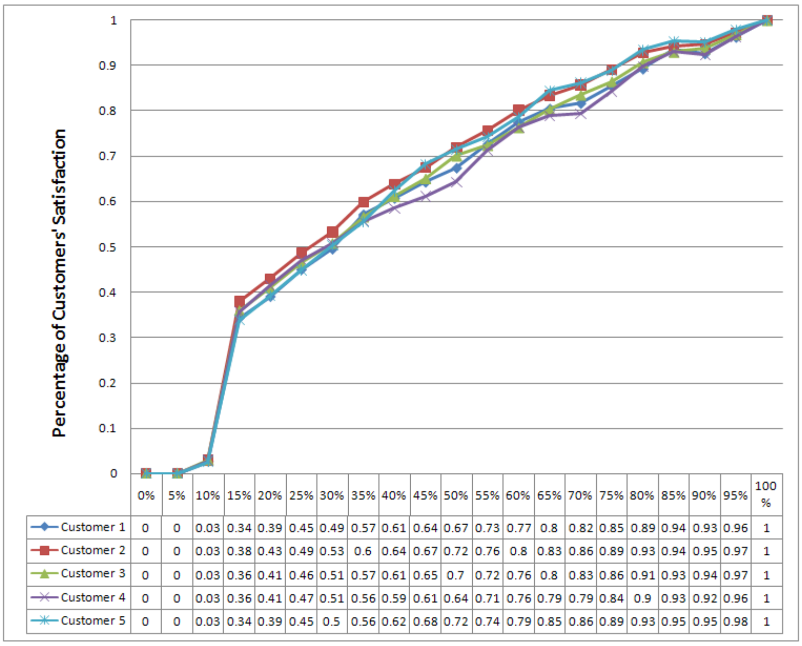
(**c**) Greer and Ruhe Data Set: 5 stake holders; 20 requirements
70% resource limitation

Figure 3. Kiviat diagrams for illustrative budget values for Greer and Ruhe Data Set



	0%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100 %
Customer 1	0	0	0.37	0.56	0.61	0.71	0.76	0.77	0.73	0.77	0.8	0.76	0.81	0.84	0.88	0.89	0.88	0.92	0.94	0.97	1
Customer 2	0	0.01	0.41	0.54	0.58	0.61	0.65	0.66	0.71	0.76	0.79	0.74	0.77	0.81	0.81	0.81	0.91	0.96	0.95	0.95	1
Customer 3	0	0	0.25	0.39	0.47	0.51	0.56	0.56	0.62	0.65	0.7	0.78	0.8	0.83	0.87	0.9	0.9	0.91	0.95	0.95	1
Customer 4	0	0.01	0.48	0.67	0.68	0.74	0.75	0.76	0.73	0.77	0.78	0.78	0.79	0.8	0.84	0.86	0.84	0.88	0.99	0.89	1

(**a**) Motorola Data Set



	0%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100 %
Customer 1	0	0	0.03	0.34	0.39	0.45	0.49	0.57	0.61	0.64	0.67	0.73	0.77	0.8	0.82	0.85	0.89	0.94	0.93	0.96	1
Customer 2	0	0	0.03	0.38	0.43	0.49	0.53	0.6	0.64	0.67	0.72	0.76	0.8	0.83	0.86	0.89	0.93	0.94	0.95	0.97	1
Customer 3	0	0	0.03	0.36	0.41	0.46	0.51	0.57	0.61	0.65	0.7	0.72	0.76	0.8	0.83	0.86	0.91	0.93	0.94	0.97	1
Customer 4	0	0	0.03	0.36	0.41	0.47	0.51	0.56	0.59	0.61	0.64	0.71	0.76	0.79	0.79	0.84	0.9	0.93	0.92	0.96	1
Customer 5	0	0	0.03	0.34	0.39	0.45	0.5	0.56	0.62	0.68	0.72	0.74	0.79	0.85	0.86	0.89	0.93	0.95	0.95	0.98	1

(**b**) Greer and Ruhe Data Set

Figure 4. Tensions between the Stake holders' Satisfaction for Different Budgetary Resource Constraints

Table 5. ANOVA analysis by multiple comparisons (Least Significant Difference) for Greer and Ruhe data set

		Diversity		Convergence	
Algorithm (x)	Algorithm (y)	Diff $(x - y)$	Sig.	Diff $(x - y)$	Sig.
5 stake holders, 20 requirements, 30% resources					
Random	Two-Archive	.8567(*)	.000	.1370(*)	.000
	NSGA-II	.0567	.458	.1370(*)	.000
Two-Archive	Random	-.8567(*)	.000	-.1370(*)	.000
	NSGA-II	-.8000(*)	.000	.0000	1.000
NSGA-II	Random	-.0567	.458	-.1370(*)	.000
	Two-Archive	.8000(*)	.000	.0000	1.000
5 stake holders, 20 requirements, 50% resources					
Random	Two-Archive	.0071	.324	.0751(*)	.000
	NSGA-II	-.0072	.317	.0751(*)	.000
Two-Archive	Random	-.0071	.324	-.0751(*)	.000
	NSGA-II	-.0143(*)	.049	-.0000	.999
NSGA-II	Random	.0072	.317	-.0751(*)	.000
	Two-Archive	.0143(*)	.049	.0000	.999
5 stake holders, 20 requirements, 70% resources					
Random	Two-Archive	.2983(*)	.000	.0540(*)	.000
	NSGA-II	.1382	.073	.0537(*)	.000
Two-Archive	Random	-.2983(*)	.000	-.0540(*)	.000
	NSGA-II	-.1600(*)	.039	-.0003	.907
NSGA-II	Random	-.1382	.073	-.0537(*)	.000
	Two-Archive	.1600(*)	.039	.0003	.907
(*) The mean difference is significant at the .05 level.					

Table 6. ANOVA analysis by multiple comparisons (Least Significant Difference) for Motorola data set

		Diversity		Convergence	
Algorithm (x)	Algorithm (y)	Mean Diff $(x-y)$	Sig.	Mean Diff $(x-y)$	Sig.
4 stake holders, 35 requirements, 30% resources					
Random	Two-Archive	.1466	.000	.1819(*)	.000
	NSGA-II	.0371	.197	.1678(*)	.000
Two-Archive	Random	-.1466	.000	-.1819(*)	.000
	NSGA-II	-.1095	.000	-.0142(*)	.001
NSGA-II	Random	-.0371	.197	-.1678(*)	.000
	Two-Archive	.1095	.000	.0142(*)	.001
4 stake holders, 35 requirements, 50% resources					
Random	Two-Archive	.0655(*)	.005	.1676(*)	.000
	NSGA-II	0.0426	.066	.1481(*)	.000
Two-Archive	Random	-.0655(*)	.005	-.1676(*)	.000
	NSGA-II	-.0229	.316	-.0194(*)	.000
NSGA-II	Random	-.0426	.066	-.1481(*)	.000
	Two-Archive	.0229	.316	.0194(*)	.000
4 stake holders, 35 requirements, 70% resources					
Random	Two-Archive	.0596	.088	.1949(*)	.000
	NSGA-II	-.0758(*)	.031	.1820(*)	.000
Two-Archive	Random	-.0596	.088	-.1949(*)	.000
	NSGA-II	-.1354(*)	.000	-.0129(*)	.000
NSGA-II	Random	.0758(*)	.031	-.1820(*)	.000
	Two-Archive	.1354(*)	.000	.0129(*)	.000
(*) The mean difference is significant at the .05 level.					

SPSS. A univariate General Linear Model with fixed factors and dependence variables was used.

Table 5 and Table 6 present the results of this analysis for the Greer and Ruhe data set and the Motorola data set respectively. The table reports the pairwise comparisons for diversity and convergence respectively. For each pair of algorithms, the mean difference between the results of applying the algorithms is presented. If the significance is smaller than .05, the difference between the algorithms is statistically significant at the 95% α level. The results show that the diversity of the Two-archive algorithm is significant in most cases. The convergence performance among Two-archive, NSGA-II and Random Search is significant. The Two-archive and NSGA-II algorithms always have a better convergence than the Random Search in the all three situations. The Two-archive and NSGA-II algorithms have almost identical performance in terms of convergence.

The analysis results for the Motorola data set presented in Table 6 shows, however, that the diversity of the three algorithms is significant in some cases. Their performance in convergence are significant, not only between metaheuristic techniques and Random Search but also between Two-Archive and NSGA-II. The Two-Archive algorithm outperforms NSGA-II and Random Search in the three instances of resource limitations. If we increase the number of objectives further, the significance increases.

6. Related Work

Several related studies have been proposed for requirements analysis and optimisation. Karlsson [12, 13, 16] proposed the Analytic Hierarchy Process (AHP) for assigning priorities to requirements and developing strategies for selecting an optimal set of requirements for implementation. The Focal Point tool (marketed by IBM) is based on this work. Our work is concerned with the requirements selection problem rather than the prioritisation problem, but, like Karlsson, we use techniques for the innovative optimisation.

Bagnall et al. [1] suggested the term *Next Release Problem* (NRP) for requirements release planning and described the various metaheuristic optimisation algorithms, including greedy, branch and bound, simulated annealing and hill climbing. 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 stake holders, whose requirements are to be satisfied. The objective was to maximise the cumulative measure of the stake holder's importance to the company under resource constraints. Feather and Menzies [2] proposed a Defect Detection and Prevention (DDP) process based on a real-world instance: a NASA pilot study. It is an iterative model to seek the near-optimal attainment of requirements by applying simulated annealing.

Ruhe et al. [3, 36] proposed the genetic algorithm based approaches known as EVOLVE family which aimed to maximise the benefits of delivering requirements in incremental software release planning process. Van den Akker et al. [37, 38] developed an optimisation 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.

The requirements analysis problem can be treated in the single objective (as described above) and multi-objective formulations, in which the various goals and constraints are converted into objective fitness functions.

The present paper adopts a multi-objective approach to requirements optimisation. Recently, several approachers have been proposed for multi-objective formulations of release planning requirements problems [7, 39]. In this work on the Multi-Objective Next Release Problem (MONRP), each of the objectives to be optimised is treated as a separate goal in its own right; the multiple objectives are not combined into a single (weighted) objective function. This allows the optimisation algorithm to explore the Pareto front of non dominated solutions. Each of these non dominated solutions denotes a possible assignment of requirements that maximises all objectives without compromising on the maximisation of the others.

Hitherto, the work on the MONRP has considered three possible multi-objective formulations, one in which the two objectives to be optimised are cost and benefit [7], one in which the two objectives are implementation-based and business-based [39], and one in which various notions of 'fairness' are the objectives [9, 23]. Compared to this work, the fairness analysis study provided a multi-objective search-based approach, in which a requirements engineer can define several possible models of fairness. Each notion of fairness forms an objective, no matter how many stake holders are involved. In the 'fairness' approach, the number of objectives is equal to the number of definitions of fairness that the decision maker wishes to analyse simultaneously. For the approach proposed in the present paper, each stake holder's satisfaction is considered to be a separate objective. The number of objectives is therefore equal to the number of stake holders and there is no concept of fairness; merely of the tension between each customer's requirements within a given budget.

While these are sensible choices, there has been no work on the most natural choice of objectives; the one in which each stake holder's requirements are considered as an objective in their own right. In this formulation, there may be many stake holders, each with their own requirements and therefore an arbitrary number of objectives.

Using Pareto optimal search it becomes possible to explore precisely the *extent* to which it is possible to satisfy "all of the people all of the time". Of course, this is unlikely to be completely achievable. The algorithm, however, attempts to produce a set of non dominated solutions that are as close as the stake holders' prioritisation will allow to this ideal situation.

In the present paper, we consider, for the first time, each stake holder as a separate and independent objective. The problem is thus characterised as a multi-objective optimisation problem.

Nuseibeh et al. [40] considered the problems associated with multiple stake holders with completing and conflicting view points. Pardee [41] presented how to use Quality Function Deployment (QFD) to satisfy and delight the stake holders. Boehm et al. [42] proposed the WinWin model to help stake holders negotiation processes based on a Multi-Criteria preference analysis. Another approach to resolve stake holder conflicts is the ViewPoint approach [43, 44] which separates the different opinions among the stake holders and can detect conflicts automatically. van Lamsweerde et al. [45] proposed goal-oriented KAOS methodology to manage conflicts and inconsistencies in RE process. In the stake holder analysis problem, Robinson et al. [46, 47] worked on the requirements negotiation model which provided automated support to generate requirements resolutions.

In the context of this related work, the present paper contains several novel approaches. It is the first to use a Pareto optimal approach to model each stake holder as a separate requirement. This is important because it treats all stake holders/stake holders as equal. Of all previous work on search-based requirements, the present paper is also the first to use visualisation to explore the results of search-based requirements analysis and, of all work on automated requirement analysis it is, to the authors knowledge, the first to use animation as a decision support tool to illustrate the results. This is important, because it frees up the decision maker, often a non-technical business analyst, from technical details of the automated optimisation process.

7. Threats To Validity and Limitations

Traditionally, in Empirical Software Engineering work there are four threats to validity that may affect the results of the empirical study presented. It is important to consider these threats to validity so that the work is contained within a complete and full understanding of its limitations. In any empirical work, the results can only be extrapolated with extreme care and it is the duty of the researcher to draw the attention of the reader to any potential issues that might affect the degree to which the results can be relied upon, extended, extrapolated or replicated.

Four threats to potential validity of findings are considered and discussed in this section: construct validity, internal and external validity and conclusion validity.

Construct validity occurs when the study uses measures and scales that can be argued to directly measure or have a strong correlation to the properties of the entity under investigation. This can be a highly controversial topic in its own right (for example the widespread controversy of using IQ to measure intelligence). In this study, the objects studied are sets of requirements and their properties of interest are the value and cost 'measurements' associated with each requirement. In our real world case study data set from Motorola, these measurements upon which we base the optimisation come from estimates, leading to one possible construct validity issue.

In this paper, the objects studied are sets of requirements and their properties. Their properties of interest are the *value* and *cost* 'measurements' associated with each requirements. These measurements on which we base the optimisation come from estimates. It is common for software engineers to rely on estimates and it is helpful in the provision of decision support to use these estimates to explore trade-offs of requirements selection. Indeed, the results of the analysis rely on the quality of the estimates. However, this might lead to one possible construct validity issue when these estimates are unreliable. Another issue is that not all the projects have clearly visible stake holders who evaluate existing requirements.

External validity, sometimes referred to as selection validity, is the degree to which the findings can be generalised. In this study two kinds of data were used in the empirical study. In order to explore the behaviour of the algorithms for different sizes of problem, synthetic data sets were created to explore performance. These data sets clearly cannot be said to be representative of real world data sets and are only used to test the performance of the algorithms used for search-based optimisation. The "27 combination random data sets" were generated to provide a better coverage of potential instances that remained uncaptured by the real world case studies. However, these data sets still cannot be said to be representative. There is missing information in the "27 combination random data sets" model. Such as, only uniform distribution is adopted in the generation of random numbers. In the real world, the data structure might be followed by other probability distributions. In addition, the different types of requirement relationships may also have different dependency density levels. All of these factors are likely to impact on the results obtained.

The real world data sets used in the paper come from Motorola and from a previous paper by Greer and Ruhe [3]. These data sets are more realistic since they come from real projects with their associated requirements. However, care is required in extrapolating from these results to the more wider context of requirements applications in general. There is a great variety of requirement analysis scenarios and so particular attention will be required in follow up studies to widen the range of data sets used.

Our approach is also limited to multi-stakeholder requirements scenarios with relatively few stake holders. Once the real world scenario has more than about 10 stake holders, the visualisations we propose and the animations we introduce will become hard to read. Furthermore, the search-based optimisation techniques we use do not, presently, scale well beyond four or five objectives. Nevertheless, we believe even approaches that can handle up to five stake holders may have value. For example, in some situations, the 'stake holders' involved, are not individual end users, but are organisations which denote parties to the choice of system requirements. Indeed this was precisely the situation with the Motorola data set used in this paper; the 'stake holders' are not individuals, they are telecommunication service providers, each with their own views of the priority of requirements.

Evolutionary multi-criteria optimisation has traditionally concentrated on problems comprising two or three objectives. Our formulation comprises a larger number of objectives in order to make it more applicable to multi-stakeholder scenarios. This is a topic of current investigation in the Search Based Optimisation community. For example, Corne and Knowles [48] compare a number of ranking methods to address the shortcoming of existing evolutionary algorithms for many-objective optimisation, while Deb and Kumar [49] suggest an interactive method to incorporate user preferences in guiding the multi-objective search.

Fortunately, even in situations where the real world scenario demands models that account for many more stake holders/stake holders than five, it will remain possible to use the approach introduced in this paper to explore tradeoffs between important subsets of stake holders and to explore tradeoffs between clusters of stake holders.

Internal validity is the degree to which conclusions can be drawn about the causal effect of the independent variable on the dependent variable. In the case of this study, the primary comparison tests for statistically significant results concerned the relative performance of the two algorithms; the two archive and NSGA-II, and also their performance against a random search. The statistical tests used in the paper make no assumptions about underlying distribution of results (making them more robust). The dependent variables measured are diversity and convergence (two widely used measures of the effectiveness of Pareto optimal optimisation algorithms).

The significance level chosen was the 95% level, denoting our balance of concerns with respect to type I and type II statistical errors. This is also the choice adopted in most other research on Empirical Software Engineering. It is a sensible balance between the two types of error for research such as this, where adoption of a technique can improve business performance (we hope) but where safety critical issues are not direct consequences of any such adoption. That is, one does not want to abandon potentially valuable new techniques by setting the alpha level to a more stringent requirement, for instance 99%, as might be suitable for trials of potential harmful drugs. However, one does want to ensure that there is a relatively robust assessment that the evidence points to a causal effect of the dependent variable on the independent variable.

Although identification of the best algorithms for the problem could be a separate subject, we feel the chosen algorithms are appropriate to address the main research questions in the studies. Our results have demonstrated that the evolutionary optimisation algorithms are superior to Random Search. While this is a relatively low threshold to aim for, it does provide a validity check on the approach. Also, in the absence of any alternative technique, Random Search is a natural choice for a base line validity check.

Conclusion validity is the degree to which conclusions reached can be said to be reasonable given the results. In the case of this paper, the work is the first to introduce a search-based approach to balance multi-stakeholder constraints and so the primary conclusion is that the evidence suggests that this approach is feasible.

This conclusion is based upon the study of real world and synthetic data sets, and provides initial evidence to support the claim, but this is merely initial evidence. The conclusion is necessarily limited by the evidence we are able to offer. Specifically, it would be too strong to claim that the approach will save requirements engineers' time or that it will help to ensure that the results of requirements analysis please the stake holders. In order to even consider such claims as these, further field study will be required. However, we believe that the modest claim that the approach is feasible is reasonable given the evidence. Also, we believe that claims relating to the relative performance of the two algorithms are reasonable with respect to the available evidence.

There are two different kinds of data set used in this work: synthetic and real world. They play different roles in our study. The synthetic data sets permit experimental evaluation, in which the parameters that affect the outcomes of the algorithms can be controlled in a laboratory setting. This is valuable for assessing how the algorithms behave in such controlled settings. However, on their own, these results can be misleading, because we have no idea whether these forms of setting arise in practice. The real world data sets serve this second role; they allow us to explore the effects of the algorithms in a real, uncontrolled, settings.

Naturally, synthetic data sets can be constructed 'to order' and so there is no limit to the amount of data we can consider. However, real world data is scarce because organisations are reluctant to let real world data out of the organisation. The degree to which results from real world data sets can be generalised is limited, becuase the data is, by definition, pertinent to a specific real world scenario. Synthetic data admits greater generalisation, but it is not clear whether this generalisation leads to observations that may be made in practice.

As such, there is no ideal form of data source; both synthetic and real world data sets give only a partial picture. By including results from a set of sets of synthetic data and two real world case studies we seek to try to combine the advantages of these two very different kinds of data, allowing us to form a sense of how the algorithms behave both in the real world and also in controlled laboratory settings.

8. Conclusions and Future Work

This paper presented the concept of internal tensioning among multi-stakeholder in requirements analysis and optimisation for the first time. A new formulation of Multi-Objective Next Release Problem was proposed, treating each stake holder as a separate objective to maximise the requirement satisfaction of each stake holder. The paper provides a simple visual method to show the tensioning of the relationships between the stake holders and resources as well as the internal tensioning among the stake holders.

The paper compares the performance of the Two-Archive algorithm, the NSGA-II algorithm and a Random Search. These three algorithms were compared according to a set of convergence and diversity metrics on a set of scalable testing data sets including both those synthetically generated to provide limiting-case data and real world industrial data. The results reveal that the Two-Archive algorithm outperformed the others in convergence as the scale of problems increase.

We envisage that the animated Kiviat diagrams can be used by a decision maker to explore the effect of budget pressure on multiple stake holder satisfaction. The decision maker might, for example, use these diagrams to help decide upon the position to adopt in external negotiations with stake holders. The requirements engineer can also see where modest increases in budget can results in significant increases in satisfaction. This analysis may be useful in internal negotiations over budgetary allowance for the project. Though the paper has presented results for real world requirements data sets, more work is required to explore these potential applications of this work.

Future work will also consider the ways in which the approach can be scaled up to handle scenarios with many more stake holders than that considered here. In this work, automated optimising clustering techniques will be considered as a way to identify groups of stake holders with similar goals. Such as approach can be used in combination with the multi-stakeholder optimisation approach introduced in this paper to explore the ways in which different groups of stake holders might interact, compete and collaborate (facilitated by automated decision support) to agree upon the most mutually agreeable set of requirements.

References

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

[2] M. S. Feather, T. Menzies, Converging on the Optimal Attainment of Requirements, in: Proceedings of the 10th IEEE International Conference on Requirements Engineering (RE '02), IEEE, Essen, Germany, 2002, pp. 263–270.

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

[4] O. Jalali, T. Menzies, M. Feather, Optimizing Requirements Decisions With KEYS, in: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering (PROMISE '08), ACM, Leipzig, Germany, 2008, pp. 79–86.

[5] J. Karlsson, C. Wohlin, B. Regnell, An evaluation of methods for priorizing software requirements, Information and Software Technology 39 (1998) 939–947.

[6] M. O. Saliu, 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, ACM, Dubrovnik, Croatia, 2007, pp. 105–114.

[7] Y. Zhang, M. Harman, S. A. Mansouri, The Multi-Objective Next Release Problem, in: Proceedings of the 9th annual Conference on Genetic and Evolutionary Computation (GECCO '07), ACM, London, UK, 2007, pp. 1129–1137 (Best Paper Award).

[8] Y. Zhang, A. Finkelstein, 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*, Springer, Montpellier, France, 2008, pp. 88–94.

[9] A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, Y. Zhang, "Fairness Analysis" in Requirements Assignments, in: Proceedings of the 16th IEEE International Requirements Engineering Conference (RE '08), IEEE Computer Society, Barcelona, Catalunya, Spain, 2008, pp. 115–124.

[10] B. Cheng, J. Atlee, From State of the Art to the Future of Requirements Engineering, in: L. Briand, A. Wolf (Eds.), Future of Software Engineering 2007, IEEE Computer Society Press, Los Alamitos, California, USA, 2007. This volume.

[11] P. Carlshamre, Release Planning in Market-Driven Software Product Development: Provoking an Understanding, Requirements Engineering 7 (2002) 139–151.

[12] J. Karlsson, Software Requirements Prioritizing, in: Proceedings of the Second International Conference on Requirements Engineering (RE '96), IEEE Computer Society, Colorado Springs, CO, USA, 1996, pp. 110–116.

[13] J. Karlsson, K. Ryan, Supporting the Selection of Software Requirements, in: Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD '96), IEEE Computer Society, Washington, DC, USA, 1996, p. 146.

[14] L. Karlsson, B. Regnell, T. Thelin, Case Studies in Process Improvement through Retrospective Analysis of Release Planning Decisions, International Journal of Software Engineering and Knowledge Engineering 16 (2006).

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

[16] K. Ryan, J. Karlsson, Prioritizing Software Requirements in an Industrial Setting, in: ICSE, pp. 564–565.

[17] M. O. Saliu, G. Ruhe, Software release planning for evolving systems, ISSE 1 (2005) 189–204.

[18] H. Kellerer, U. Pferschy, D. Pisinger, Knapsac Problems, Springer, 2004.

[19] M. Harman, The Current State and Future of Search Based Software Engineering, in: 29th Int. Conference on Software Engineering (ICSE 2007), Future of Software Engineering (FoSE), Minneapolis, USA.

[20] J. T. de Souza, C. L. Maia, F. Gomes, de Freitas, D. P. Coutinho, The Human Competitiveness of Search Based Software Engineering, in: Proceedings of 2^{nd} International Symposium on Search based Software Engineering (SSBSE 2010), IEEE Computer Society Press, Benevento, Italy, 2010, p. To Appear.

[21] K. W. Kolence, The Software Empiricist, ACM SIGMETRICS Performance Evaluation Review 2 (1973) 31–36.

[22] K. W. Kolence, P. J. Kiviat, Software Unit Profiles & Kiviat Figures, ACM SIGMETRICS Performance Evaluation Review 2 (1973) 2–12.

[23] A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, Y. Zhang, A Search based Approach to Fairness Analysis in Requirement Assignments to Aid Negotiation, Mediation and Decision Making, Requirements Engineering Journal (RE '08 Special Issue) 14 (2009) 231–245.

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

[25] N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation 2 (1994) 221–248.

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

[27] J. Horn, N. Nafpliotis, Multiobjective Optimization Using The Niched Pareto Genetic Algorithm, Technical Report IllIGAL 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, IL, 1993.

[28] K. Praditwong, X. Yao, A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm, in: Proceedings of he 2006 International Conference on Computational Intelligence and Security (CIS'2006), IEEE Press, 2006, pp. 286–291.

[29] J. D. Knowles, D. W. Corne, M. J. Oates, The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization, in: Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI), Springer, Berlin, 2000, pp. 839–848.

[30] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[31] Y. Zhang, E. Alba, J. J. Durillo, S. Eldh, M. Harman, Today/Future Importance Analysis, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10), ACM, Portland, USA, 2010. To appear.

[32] P. Baker, M. Harman, K. Steinhöfel, 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), IEEE Computer Society, Philadelphia, Pennsylvania, 2006, pp. 176–185.

[33] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, 2001.

[34] K. Praditwong, X. Yao, A New Multi-Objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm, in: Proceedings of he 2006 International Conference on Computational Intelligence and Security (CIS '06), volume 1, IEEE Press, Guangzhou, China, 2006, pp. 286–291.

[35] I. SPSS, http://www.spss.com/, ????

[36] G. Ruhe, A. Ngo-The, Hybrid Intelligence in Software Release Planning, International Journal of Hybrid Intelligent Systems 1 (2004) 99–110.

[37] C. Li, M. van den Akker, S. Brinkkemper, 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*, Springer, Trondheim, Norway, 2007, pp. 93–108.

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

[39] M. O. Saliu, G. Ruhe, Bi-Objective Release Planning for Evolving Software Systems, in: Proceedings of ESEC/SIGSOFT FSE 2007.

[40] B. Nuseibeh, J. Kramer, A. Finkelstein, A Framework for Expressing the Relationships between Multiple Views in Requirements Specifications, IEEE Transactions on Software Engineering 20 (1994) 760–773.

[41] W. J. Pardee, To Satisfy & Delight Your Customer: How to Manage for Customer Value, Dorset House Publishing Company, 1996.

[42] B. Boehm, P. Bose, E. Horowitz, M. J. Lee, Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach, in: Proceedings of the 17th International Conference on Software Engineering (ICSE '95), ACM, Seattle, Washington, US, 1995, pp. 243–253.

[43] S. Easterbrook, M. Chechik, A Framework for Multi-Valued Reasoning Over Inconsistent Viewpoints, in: Proceedings of the 23rd International Conference on Software Engineering, IEEE Computer Society Press, Toronto, Ontario, Canada, 2001, pp. 411–420.

[44] S. Easterbrook, A. Finkelstein, J. Kramer, B. Nuseibeh, Co-Ordinating Distributed Viewpoints: The Anatomy of a Consistency Check, Concurrent Engineering 2 (1994) 209–222.

[45] A. van Lamsweerde, R. Darimont, E. Letier, Managing Conflicts in Goal-Driven Requirements Engineering, IEEE Transactions on Software Engineering 24 (1998) 908–926.

[46] W. Robinson, S. Fickas, Automated Support for Requirements Negotiation, 1994.

[47] W. N. Robinson, V. Volkov, Requirement Conflict Restructuring, 1999.

[48] D. Corne, J. Knowles, Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others, in: GECCO'07: Proceedings of Genetic and Evolutionary Computation Conference, pp. 773–780.

[49] K. Deb, A. Kumar, Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method, in: GECCO'07: Proceedings of Genetic and Evolutionary Computation Conference, pp. 781–788.