Requirements Prioritization and Next-Release Problem under Non-Additive Value Conditions

Ashish Sureka
Indraprastha Institute of Information Technology, Delhi (IIITD)
New Delhi (India)
ashish@iiitd.ac.in

Abstract—Next Release Problem (NRP) is a complex combinatorial optimization problem consisting of identifying a subset of software requirements maximizing the business value under given constraints such as cost and resource limitation, time and functionality related dependencies between requirements. NRP can be mathematically formulated as an integer linear programming problem and previous researches solve the NRP multi-objective optimization problem using exact and metaheuristic search techniques. We present a mathematical formulation of the NRP under conditions of non-additive customer valuations (positive and negative synergies) across requirements. We present a model that allows customers to state their preferences or valuations across bundles or combinations of requirements. We analyze the economic efficiency gains, cognitive and computational complexity of the proposed model. We conduct experiments to investigate the applicability of multi-objective evolutionary algorithms (MOEAs) in solving the NRP with non-additive valuations and implication constraints on requirements. We compare and contrast the performance of state-of-the-art MOEAs such as NSGA-II and GDE3 on synthetic dataset representing multiple problem characteristics and size and present the result of our empirical analysis.

Keywords—Next Release Problem, Requirements Management, Feature Selection Problem, Multi-Objective Optimization, Value-Based Requirements Interdependency, Metaheuristic Search, Evolutionary Algorithms, Search-Based Software Engineering

I. RESEARCH MOTIVATION AND AIM

Next Release Problem (NRP) or Feature Subset Selection (FSS) problem is defined as a decision problem consisting of identifying the software features to be implemented for the next release based on several factors such as the value of the feature to the customer(s), cost of implementing each feature by the organization, dependencies between features, relative importance of the customer, budget and constraints related to scheduling and available expertise [1] [5] [6] [7] [8] [11] [12]. NRP is a complex optimization problem (an NP Hard Problem which is shown to be equivalence to the Knapsack problem) as the number of feasible solutions grows exponentially with the number of requirements. The size of the solution space is \( 2^N \) for \( N \) requirements (\( N \) decision variables in which the value of a decision variable is either 0 or 1) and for large problems it becomes computationally expensive or practically infeasible to conduct an exhaustive search to find an optimal (or pareto-optimal solution set) solution. Solving the NRP using Integer Linear Programming (ILP), exact techniques and meta-heuristics search techniques is an area that has attracted several researcher’s attention [1] [5] [6] [7] [8] [11] [12]. Previous approaches model and solve the NRP such that the requirements are independent from each other from the perspective of business value to the customer (and not in-terms of sequential dependency or time-based dependency). Based on our analysis of previous work on NRP, we observe that while value-based requirements interdependencies is mentioned [2] [3] [8], there is no focused study on formulating and solving the NRP (exact or approximate techniques) under conditions on positive and negative synergy between requirements across all unique requirement bundles or packages.

Additive valuation means that if the value for the requirements \( R_x \) and \( R_y \) for the customer is \( V(R_x) \) and \( V(R_y) \) respectively then the valuation for the requirements \( V(R_x, R_y) \) is \( V(R_x) + V(R_y) \). However, in reality the valuations for requirements may not always be additive. The valuations for the requirements can be complementary or having positive synergy: \( V(R_x, R_y) \) is greater than \( V(R_x) + V(R_y) \). Similarly, the valuation of the requirements can have substitutability property or negative synergy: \( V(R_x, R_y) \) is less than \( V(R_x) + V(R_y) \). Positive and negative synergy between requirements results in non-additive valuations. Capturing valuations of customers over combinations of requirements (bundling or packages) rather than individual requirements results in more efficient allocation of resources (refer to the literature on Combinatorial Auctions [4] [9] [10]) in comparison to the model in which the valuations are additive. The study presented in this paper is motivated by the need to investigate the Next Release Problem under non-additive valuation conditions. The specific research aim of the work presented in this paper is the following:

1) To extend the mathematical formulation (an integer linear program and an NP hard optimization problem) of the NRP by incorporating non-additive customer valuations (positive and negative synergy between requirements)

2) To demonstrate economic efficiency gains under non-additive valuation conditions in comparison to additive valuations assumption for the NRP.

3) To investigate the application of (MOEA) for solving the complex next-release optimization problem under non-additive valuation conditions. To compare and contrast the performance of state-of-the-art MOEA on a variety of synthetic problems of varying characteristics.
II. RELATED WORK AND RESEARCH CONTRIBUTIONS

In this Section, we present closely related work to the study presented in this paper and list novel research contributions in context to existing work.

A. Requirements Interdependencies

Dahlstedt et al. provide an overview of the state of research on requirements interdependencies and formulate a research agenda, unresolved issues and research challenges for the area of requirement interdependencies in the software development process [3]. Li et al. present six types of requirement dependencies and list natural combination of various dependencies (such as time-related dependencies and implication dependency) [8]. Carlshamre et al. conduct a survey of 5 distinct products from 5 different companies, defines different categories of dependencies between requirements and present a technique for visualization of interdependencies with the aim of supporting release planning [2].

B. Exact Techniques for Solving Next Release Problem

Li et al. present an integer programming model that integrates time scheduling into software release planning and their model incorporates requirement scheduling in addition to requirement selection. They use ILOG CPLEX for solving the integer linear programming problems [8]. Akker et al. model several management steering mechanism in software release planning. They use two ILP software packages: Solver included in Microsoft Excel professional edition and ILOG CPLEX [11]. Jung apply the integer programming solver to solve two real-world example [7].

C. Meta-heuristic Search Techniques for Solving Next Release Problem

Bagnall et al. employ hill climbing algorithms and simulated annealing for solving the next release problem [1]. Greer describe an evolutionary and iterative approach called as EVOLVE and apply genetic algorithms to generate optimal or near-optimal solution to solve the next release problem [6]. Freitas et al. compare and contrast exact techniques (simplex method branch-and-bound), genetic algorithm and simulated annealing based approach [5]. The study conducted by Zhang et al. provides evidences to support the claim that weighted and Pareto optimal genetic algorithms together with the NSGA-II are well suited to solve the multi-objective next release problem [12]. In context to existing work, the study present in this paper makes the following novel research contributions:

1) An integer linear programming formulation of the next release problem incorporating positive and negative synergy (value based inter-dependencies) between requirements across all combination of requirement bundles. We discuss economic efficiency gains and cognitive and computational load imposed by the proposed model (due to non-additive customer valuations).

2) A study investigating the application of multi-objective optimization evolutionary algorithms for solving the multi-objective next release problem under non-additive valuation condition. We generate several synthetic problems and apply various multi-objective optimization evolutionary algorithms, compare and contrast the performance of various algorithms and present our insights.

III. SOLUTION APPROACH

A. Value-Based Requirements Interdependencies

The focus of the study presented in this paper is on value-based requirement interdependency and incorporating positive as well as negative synergy between requirements in the next-release combinatorial optimization problem. Carlshamre et al. conduct an industrial survey on requirements interdependencies and their study reveals that only a few requirements are singular and majority of the requirements are interdependent [2]. They define value (one requirement positively or negatively affects the value of another requirement for a customer) and cost (one requirement affects the cost of implementing another requirement) based dependency in addition to temporal and functional dependency (one requirement requires another requirement to function or one requirement has to be implemented before another requirement) [2]. Dahlstedt et al. argue that requirements cannot be treated as stand-alone artefacts, cannot be selected based solely on priority and present an integrated classification of fundamental requirement interdependency types [3]. They define two broad categories of interdependencies which are fundamental and neutral: structural dependencies and cost or value dependencies [3]. Li et al. define six types of requirement dependencies: combination, implication, exclusion, revenue-based, cost-based and time-related [8]. Previous work on requirement interdependencies and next release problem discuss value-based dependencies but we believe there are lack of focused and in-depth studies on incorporating value-based requirement dependencies in the next release optimization problem. We believe that mathematically formulating the next-release problem under conditions of value-based requirement dependencies and investigating exact as well as approximate techniques for solving the complex optimization problem is an area which is relatively unexplored. In this paper, we present a model which not only incorporates positive and negative synergy between two requirements but between all combinations and bundles of requirements.

B. Example Demonstrating Economic Efficiency Gain

Consider three requirements: \( R_x, R_y \) and \( R_z \), Table I displays two scenario. The Table on the left side shows the value and cost of requirements \( R_x, R_y \) and \( R_z \) (assuming additive values i.e. \( V(R_xR_y) = V(R_x) + V(R_y) \)). The optimal solution to the requirement prioritization problem (combinatorial optimization problem) for the additive value scenario (Table I Left) is to select requirements \( R_xR_z \) (assuming the budget or cost constraint is 6) for the next release. The value of the optimal solution is 13 and the cost is 5 (maximizing value under a given cost constraint). Consider the scenario depicted on the right hand side of the Table I. As shown in Table I, the number of bundles (unique combination of requirements) is 8 (including the empty set). Table I shows the value of the customer for each of the 8 bundles. In the given scenario (focus of the work presented in this paper), the valuation for a bundle of requirement is not equal to the sum of the valuations of the individual requirements in the bundle. The value of \( R_xR_y \) is
14 but the value of $R_x$ is 5 and the value of $R_y$ is 5 (the values are non-additive i.e. $V(R_x R_y) \geq V(R_x) + V(R_y)$). The non-additive value between $R_x$ and $R_y$ shows synergy between the two requirements (which is a realistic situation in comparison to a situation with non-additive values). The value of $R_y R_z$ is 10 but the value of $R_y$ is 5 and the value of $R_z$ is 8 (the values are non-additive and is an example of negative synergy or substitutability i.e. $V(R_x R_y) \leq V(R_x) + V(R_y)$). The optimal solution to the requirement prioritization problem for the non-additive value scenario (Table I Right) is to select requirements $R_x R_y$ (assuming the budget or cost constraint is 6, maximizing value under given cost constraints) for the next release. The worked out example in Table I shows that the optimal solution under additive and non-additive valuation conditions can be different (non-additive valuation model captures reality and results in a more economic efficient solution in comparison to the additive value conditions).

### C. Optimization Problem Definition and Formulation

Let $R$ be the set the of $N$ requirements index by $i$: $R = \{R_1, R_2, ..., R_N\}$. Let $S$ denote a subset of requirements $R$ i.e., $S \subseteq R$. The number of bundles or packages (combinations of requirements) for $N$ are $2^N$ including the null set $\emptyset$. In traditional approaches (previous study) of Next Release Problem, the customers need to provide their valuation over $N$ items (the valuations are assumed to be additive). However, in the proposed approach the customer needs to provide valuation over combinations of items ($S \subseteq R$) as in the proposed approach we are assuming non-additive valuations (positive and negative synergy between requirements). The non-additive valuation assumption changes the objective function, constraints and search space. Let $CR$ be the set of $M$ customers index by $j$: $CR = \{CR_1, CR_2, ..., CR_M\}$. $V_j(S)$ represent the value of requirement bundle $S$ by customer $j$. Let $w_j$ denote the importance or weight of customer $j$ for the software vendor. The objective function $Z$ (a maximization function) is to maximize business value for the software vendor (equation 1).

$$Z = \max_{j=1}^{M} \sum_{S \subseteq R} w_j \cdot v_j(S)$$

We denote $Z^*$ to be the objective function value for the optimal solution $S^*$. We represent decision vector for the objective function as $x = x_1, x_2, ..., x_N$. $x_i$ is a binary decision variable which can take the value 0 or 1. $x_i = 1$ means that the requirement $R_i$ is chosen as part of the solution vector and $x_i = 0$ means that the requirement $R_i$ is not selected. For example, consider three requirements: $R_1, R_2$ and $R_3$. In the example of three requirements, the optimal solution $S^* = \{R_1, R_3\}$ will be denoted by the optimal decision vector $x^* = 1, 0, 1$. Let $cost_i$ denote the cost of implementing requirement $R_i$ and let $B$ be the total budget. Equation 2 is the formula for the budget constraint for the objective function defined in Equation 1.

$$\sum_{i=1}^{N} cost_i \cdot x_i \leq B$$

We model dependency constraints between requirements in the form a directed acyclic graph denoted by RDG (requirements dependency graph). RDG consists of $N$ nodes in which each node represents one of the $N$ requirements. The directed edge $E$ between two requirements $(R_i, R_j)$ denotes that requirement $R_i$ is a pre-requisite to requirement $R_j$. If $R_i$ is a pre-requisite to requirement $R_j$ then a solution containing $R_i$ but not $R_j$ is a feasible solution but a solution containing $R_j$ but not $R_i$ is an infeasible solution. RDG is an acyclic graph and the relationship between the requirements is transitive. If $R_i$ is a pre-requisite to $R_j$ and $R_j$ is a pre-requisite to $R_k$ then including $R_k$ in the solution means including $R_k$ and $R_i$ also for the solution to be called as feasible solution. The constraints can be defined using the following equation.

$$\text{If } x_i = 1 \text{ in } x \text{ then prerequisite}(x_i) = 1$$

### IV. MULTI-OBJECTIVE OPTIMIZATION USING EVOLUTIONARY ALGORITHMS

#### A. Evolution Algorithm Performance Indicators

We compute the performance of evolutionary algorithms on experimental dataset using five widely used quantitative performance measures implemented in the MOEA Framework Java Library. The five performance indicators are: Hypervolume (HV), Generational Distance (GD), Inverted Generational Distance (IGD), Additive epsilon indicator (AEI) and Maximum Pareto Front Error (MPFE). The five performance indicators measure performance both in terms of convergence and diversity. A reference set is needed for comparison with the solution set under interest for which the performance indicator value needs to be computed. The best solution obtained from multiple simulation runs and execution is used as the reference set for computing quantitative performance measures since the true Pareto optimal set is unknown for large problems (and that is why evolutionary algorithms are applied).

#### B. Experimental Results

We employ MOEA Framework\(^1\) which is a Java-based open-source and free library providing implementation for several state-of-the-art general purpose multi-objective optimization algorithms. We conduct experiments using four algorithms: NSGA-II, eMOEA, GDE3 and MOEA/D. We generate a synthetic dataset consisting of two problems: 50 requirements with additive value for all combination of requirements and 50 requirements with non-additive values for requirement packages. Figure shows the approximation set obtained after applying four evolutionary algorithms on two problems (additive

\(^1\)http://www.moeaframework.org/
and non-additive values) consisting of 5000 objective function evaluation runs. Experimental results in Figure 1 and 3 reveals the relative performance of four algorithms. Figure 2 and 4 presents the run-time dynamics of the behavior of εMOEA algorithm throughout the duration of a run displaying how the solution quality changes. Figure 2 and 4 indicates that speed of convergence and the gradual level-off in objective function value after 4000 objective function evaluation runs. Table II displays the minimum, median and maximum values of the five performance indicators of four evolutionary algorithm runs on two problems.

V. CONCLUSIONS

We present a mathematical formulation of the Next Release Problem under conditions of non-additive customer valuations across requirements. We conduct empirical analysis to examine the applicability of MOEAs in solving the NRP and compare the performance of state-of-the-art MOEAs on synthetic dataset.

VI. ACKNOWLEDGEMENTS

The research study presented in this paper is supported by DST FAST (Government of India) grant awarded to the author.

REFERENCES