Technical Report

Complexity Analysis for the NRP Backbone

Jifeng Xuan, He Jiang, Zhilei Ren
E-mail: xuan@mail.dlut.edu.cn

School of Software, Dalian University of Technology, China

1. Introduction

In this report, we will introduce the complexity analysis of the Next Release Problem (NRP) backbone. This report is the supplement material for the paper about the Backbone based Multilevel Algorithm (BMA) for the Next Release Problem (NRP) [2].

The NRP is a combinatorial optimization problem in search based requirements engineering. The problem model can be found in [1]. The definition of the NRP can be found in our paper of BMA.

2. Definition of the NRP

In this section, we present the related definitions of the NRP.

The NRP can be retrieved from the following scenario [1]: in the requirements analysis phase of a software project, a necessary step is to select adequate requirements in the next release to achieve maximized commercial profits within a limited cost. Each customer requests a fraction of those candidate requirements and provides a potential commercial profit for the software company. In a real-world project, the dependencies among candidate requirements restrict the selection of customer profits. The NRP aims to determine a subset of customers to achieve maximum profits under a predefined budget bound.

According to this application scenario, we give the formal definitions of the NRP as follows. In a software project, let R be the set of all the candidate requirements and the cardinality of R is $|R| = m$. Each requirement $r_j \in R$ $(1 \le j \le m)$ is associated with a nonnegative cost $c_j \in C$. A directed acyclic graph $G = (R, E)$ denotes the dependencies among these requirements, where R is the set of vertexes and E is the set of arcs. In the dependency graph G, an arc $(r', r) \in E$ indicates that the requirement r depends on r', i.e., if r is implemented in the next release, r' must be implemented as well to satisfy the dependency. Let $parents(r)$ be the set of requirements, which can reach r via one or more arcs. More formally, $parents(r) = \{r' \in R | (r', r) \in E \ or \ (r', r'') \in E, \ r'' \in parents(r)\}$. Obviously, all the requirements in $parents(r)$ must be implemented to ensure the implementation of r.

Let S be all the customers related to the requirements R and $|S| = n$. Each customer $s_i \in S$, requests a set of requirements $R_i \subseteq R$. Let $w_i \in W$ be the profit gained from the customer s_i. Let $parents(R_i) = \bigcup_{r \in R_i} parents(r)$. For a given customer s_i, let the set of total requirements requested by s_i be $\hat{R}_i = R_i \cup parents(R_i)$. Under the above definitions, a customer s_i can be satisfied by the software release decision, if and only if all the requirements in $\hat{R}_i$ are implemented in the next release. Let the cost for satisfying the customer s_i be $cost(\hat{R}_i) = \sum_{r_j \in \hat{R}_i} c_j$. A subset of customers $S_0 \subseteq S$ can be viewed as a solution. To facilitate the following discussion, we also formulate a solution as a set of ordered pairs, i.e., the solution $S_0 \subseteq S$ is denoted as $X = \{(i, p) | p = 1, s_i \in S_0 \ or \ p = 0, s_i \notin S_0\}$. It is easy to convert the form of X or S_0 into each other. Let the cost of a solution X be $cost(X) = cost(\bigcup_{(i,1) \in X} \hat{R}_i)$ and the objective function of X (i.e., the profit of X) be $\omega(X) = \sum_{(i,1) \in X} w_i$.

Definition 1. *The next release problem (NRP).*

Given a directed acyclic requirements dependency graph $G = (R, E)$, each customer $s_i \in S$ directly requests a set of requirements R_i . The profit of s_i is $w_i \in W$ and the cost of requirement $r_j \in R$ is $c_j \in C$. A predefined budget bound is b.

The goal of the NRP is to find an optimal solution X^*, to maximize $\omega(X)$, subject to $cost(X) \le b$.

For an NRP instance, the scale is $n = |S|$. To simplify our statement, all the values of an NRP instance are integers except especial specifications. For a real-world application, it is easy to convert a non-integer NRP instance into an integer-only instance by magnifying the same multiple for all the values.

From the definition of the NRP, the requirements $\hat{R}_i$ requested by a customer s_i are calculated from the dependency graph of requirements [1]. If we directly input the requirements requests for each customer, Definition 1 can be simplified [3].

Definition 2. *The Simplified NRP.*

Given a set of requirements R and a set of customers S, each requirement $r_j \in R$ $(1 \le j \le m)$ has a cost $c_j \in C$ and each customer $s_i \in S$ $(1 \le i \le n)$ has a profit $w_i \in W$. A request $q_{ij} \in Q$ shows whether a customer s_i requests a requirement r_j in the next release, i.e., $q_{ij} = 1$ denotes that s_i requests r_j or $q_{ij} = 0$ denotes not. Given a solution X, the requirements for X is $R(X) = \bigcup_{(i,1)\in X, q_{ij}=1}\{r_j\}$. A predefined budget bound is b.

The goal of the NRP is to find an optimal solution X^*, to maximize $\omega(X) = \sum_{(i,1)\in X} w_i$, subject to $cost(X) = \sum_{r_k \in R(X)} c_k \le b$.

Based on the definitions, each NRP instance can be directly converted into a Simplified NRP instance. The dependencies among requirements are included in the requirements requests Q. To simplify the following statement, a Simplified NRP instance is called an NRP instance for short. We denote an NRP instance as Π.

We define the NRP backbone in Definition 3.

Definition 3. *The NRP backbone.*

Given an NRP instance Π, let $\Gamma^* = \{X_1^*, X_2^*, \dots, X_u^*\}$ be the set of all the optimal solutions to Π. The backbone of Π is defined as $\xi = \bigcap_{i=1}^{u} X_i^*$.

The scale of ξ is $|\xi|$. Based on Definition 3, the NRP backbone contains the common characteristics of the optimal solutions. Given an NRP instance, we can reduce the instance scale by fixing its backbone.

3. Complexity Analysis for the NRP backbone

We will give the computational complexity analysis which shows that there exists no polynomial time algorithm to obtain the NRP backbone. Before giving the final analysis, we list some preliminary definitions and properties.

Given an NRP instance Π, we define its biased instance as $\hat{\Pi}$, where $\hat{W} = \{\hat{w}_i | \hat{w}_i = w_i + 1/2^i, s_i \in S\}$. In other words, a biased instance can be viewed as the original instance with noise profits. Note that the original w_i is an integer while the new $\hat{w}_i$ is not. It takes $O(n)$ running time to construct a biased instance for the given NRP instance. For a biased instance, the objective function of solution X is defined as $\hat{\omega}(X)$ accordingly. The backbone of a biased instance $\hat{\Pi}$ is denoted as $\hat{\xi}$. We call an instance Π as a unique-optimum instance, if and only if there exists exactly one optimal solution to this instance.

According to the above definitions, Property 1 and Property 2 are proposed.

Property 1. Any feasible solution to the instance Π is a feasible solution to the biased instance $\hat{\Pi}$, and vice versa.

Property 2. For the NRP instance Π and its backbone ξ, if and only if $|\xi| = n$, this instance is a unique-optimum instance.

Based on the above properties, we give the complexity analysis for obtaining the NRP backbone. It consists of three steps: first, we prove that a biased NRP instance is a unique-optimum instance; second, we analyze the relationship between an NRP instance and its biased instance; third, we prove that it is $\mathcal{NP}\text{-}hard$ to obtain the NRP backbone.

Theorem 1. *For an NRP instance Π, if w_i is an integer for any $s_i \in S$, the biased NRP instance $\hat{\Pi}$ is a unique-optimum instance.*

Proof. First, any NRP instance Π in the discussion must have its feasible solutions. Then, the biased instance $\hat{\Pi}$ has feasible solutions by Property 1. If $\hat{\Pi}$ has only one feasible solution, $\hat{\Pi}$ is a unique-optimum instance; if $\hat{\Pi}$ has two or more feasible solutions, we shall explain that the profits of such solutions must be distinct.

Given any two different feasible solutions X_a and X_b to the biased instance $\hat{\Pi}$, there must be at least one ordered pair $(i, 1)$, which only exists in one of the two solutions. In such ordered pairs, let k be the smallest i. Without loss of generality, we assume that $(k, 1) \in X_a$ and $(k, 0) \in X_b$. According to the definition of $\hat{W}$, $\hat{\omega}(X_a) - \hat{\omega}(X_b) = w_k + 1/2^k + \cdots$. The non-integer part of $\hat{\omega}(X_a) - \hat{\omega}(X_b)$ is larger than zero. Thus, $\hat{\omega}(X_a) \ne \hat{\omega}(X_b)$ holds. Therefore, all the feasible solutions to $\hat{\Pi}$ have different objective function values from each other. Thus, $\hat{\Pi}$ must have only one optimal solution. The theorem is proved. □

According to Theorem 1, we give two lemmas for further complexity analysis. Theorem 1 and these two lemmas can be also applied to non-integer instances via the instance magnification introduced in Section 2.

Lemma 1. *For an NRP instance* Π, w_i *is an integer for any* $s_i \in S$. *Given any two distinct feasible solutions* X_a *and* X_b *to an NRP instance* Π, *if* $\omega(X_a) < \omega(X_b)$, *then* $\widehat{\omega}(X_a) < \widehat{\omega}(X_b)$ *for the biased NRP instance* $\widehat{\Pi}$.

Proof. According to instance Π, $\omega(X_b) - \omega(X_a) \geq 1$. The difference of objective function values for $\widehat{\Pi}$ is

$\widehat{\omega}(X_b) - \widehat{\omega}(X_a)$
$= \omega(X_b) - \omega(X_a) + \sum_{(i,1)\in X_b} 1/2^i - \sum_{(i,1)\in X_a} 1/2^i$
$\geq \omega(X_b) - \omega(X_a) - \sum_{(i,1)\in X_a} 1/2^i$
$\geq \omega(X_b) - \omega(X_a) - \sum_{s_i\in S} 1/2^i$
$= \omega(X_b) - \omega(X_a) - (1 - 1/2^n)$
$\geq 1 - (1 - 1/2^n) > 0.$

Thus, this lemma is proved. □

Lemma 2. *For a NRP instance* Π, *if* w_i *is an integer for any* $s_i \in S$, *the optimal solution to the biased NRP instance* $\widehat{\Pi}$ *is an optimal solution to* Π.

Proof. By Property 1, the optimal solution X^* to $\widehat{\Pi}$ is a feasible solution to Π. We will prove by contradiction that X^* is an optimal solution to Π.

Assume that the optimal solution X^* to $\widehat{\Pi}$ is not an optimal solution to Π. Thus, there exists a solution X such that $\omega(X) > \omega(X^*)$. By Lemma 1, $\widehat{\omega}(X) > \widehat{\omega}(X^*)$. This contradicts with the optimal solution X^* to $\widehat{\Pi}$. Thus, Lemma 2 is proved. □

Based on the above lemmas, we give the complexity analysis for obtaining the NRP backbone.

Theorem 2. *The NRP backbone problem.*

Unless $\mathcal{P} = \mathcal{NP}$, there exists no polynomial time algorithm to obtain the NRP backbone (it is $\mathcal{NP}$-$hard$ to obtain the NRP backbone).

Proof. We prove this theorem by contradiction. Assume that this theorem is false, i.e., there must exist an algorithm Ξ, which can obtain the backbone ξ of the NRP within polynomial time (denoted as $O(f(p))$, where $f(p)$ is a polynomial function of $p = m + n + \sum_{r_j\in R}\lceil \log_2(c_j + 1)\rceil + \sum_{s_i\in S}\lceil \log_2(w_i + 1)\rceil$ and $\lceil\cdot\rceil$ denotes the maximum integer $\leq \cdot$.

Given any NRP instance Π, for any $s_i \in S$, we only consider the case that w_i is an integer based on the above two lemmas. For a non-integer instance, we can generate an integer-only instance via the instance magnification. Now we can construct an algorithm to solve Π as follows.

a) We construct the biased instance $\widehat{\Pi}$ for Π in $O(n)$ running time;

b) According to the assumption, since the biased instance $\widehat{\Pi}$ is also an instance of the NRP, its backbone $\hat{\xi}$ can be obtained within $O(f(p^2))$ running time by the algorithm Ξ;

c) By Lemma 1, $\widehat{\Pi}$ is a unique-optimum instance. Thus, $\hat{\xi}$ is an optimal solution to $\widehat{\Pi}$;

d) By Lemma 2, the optimal solution to $\widehat{\Pi}$ is also an optimal solution to Π. Thus, $\hat{\xi}$ is also an optimal solution to Π.

Therefore, the optimal solution to Π can be obtained within $O\big(f(p^2)\big) + O(n)$ running time. This contradicts with the fact that the NRP is $\mathcal{NP}$-$hard$. Thus, this theorem is proved. □

Acknowledgments

We thank our anonymous reviewers for valuable comments and corrections of this part.

References

[1] A. Bagnall, V. Rayward-Smith, and I. Whittley, "The Next Release Problem," *Information and Software Technology*, vol. 43, no. 14, pp. 883-890, Dec. 2001, doi:10.1016/S0950-5849(01)00194-X.

[2] H. Jiang, J. Xuan, and Z. Ren, "Approximate Backbone Based Multilevel Algorithm for Next Release Problem," *Proc 12th Ann. Conf. Genetic and Evolutionary Computation (GECCO '10)*, pp. 1333-1340, Jul. 2010, doi:10.1145/1830483.1830730.

[3] Y. Zhang, M. Harman, and S.A. Mansouri, "The Multi-Objective Next Release Problem," *Proc. 9th Ann. Conf. Genetic and Evolutionary Computation (GECCO '07)*, pp. 1129-1136, Jul. 2007, doi:10.1145/1276958.1277179.