The Sixth International Symposium on Operations Research and Its Applications (ISORA'06)
Xinjiang, China, August 8–12, 2006
pp. 312–318

Fuzzy Optimization Models for Quality and Cost of Software Systems Based on COTS

Xueshi Shen Yingwu Chen Lining Xing

School of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China

Abstract With the rapid increasing of the complexity, size and cost of software systems, the use of existing COTS (Commercial Off-The-Shelf) products has been playing a more and more important role in developing a high quality software system to satisfy both the user requirements and the developer budget. So selecting the best COTS software products has become the key to the quality and cost of software systems based on COTS. This paper presents a fuzzy optimization model for selecting the best COTS software product. The objective function of the model is to maximize quality within a fuzzy budgetary constraint. The software system consists of several programs, where a specific function of each program can call upon a series of modules. Several alternative COTS products are available for each module. A weight to the modules is given by utilizing the Analytic Hierarchy Process (AHP) based on the access frequencies of the modules. A simplified example is given to demonstrate each optimization model, and the results show that the proposed model is correct, feasible and valid.

Keywords quality optimization; software system; COTS products; fuzzy model

1 Introduction

With the rapid increasing of the complexity, size and cost of software systems, the use of existing COTS (Commercial Off-The-Shelf) products has been playing a more and more important role in developing a high quality software system to satisfy both the user requirements and the developer budget. With COTS products, we can achieve the lower development cost, the higher development speed and the more flexible maintenance performance, and gradually obtain the ability of adopting new technology in the lifecycle of software system. Component Based on Software Engineering (CBSE) process model has become a kind of process model of software development project [1,2], so selecting the best COTS software products has become the key to the quality and cost of software systems based on COTS. On the other hand, since software development is not an exact science, there are often plenty of indefinite and uncertain factors in its quality and cost. This paper proposes a fuzzy optimization model for selecting the best COTS software product. The objective function of the model is to maximize quality within a fuzzy budgetary constraint and the limitation of the incompatibility among COTS products.

In this paper, a simplified system prototype is given to demonstrate the fuzzy optimization model. This system prototype consists of three programs, four modules, and eleven COTS products. Some specific functions of each program can call upon a series of modules, and several alternative COTS products are available for each module. Fig.1 shows the hierarchy of our example. In software system, each module in a software system has different levels of importance that depend on access frequency, so a weight to the modules is given by utilizing the AHP (Analytic Hierarchy Process) approach based on the access frequencies. This study requires knowing the quality level and cost of each COTS product. The cost is only based on the purchasing price, whereas the quality level can be estimated using any of the quality models available such as ISO/IEC 9126 [3] or others. Currently, the ISO/IEC 9126 standard is widely used by researchers and practitioners to evaluate software quality.

The paper is organized as follows. Section 1 is the introduction. Section 2 proposes elementary hypotheses and example data for the fuzzy optimization model. In addition, Section 2 assigns weights to modules of the software system. Section 3 addresses a fuzzy optimization model for selecting the best COTS product for each module, with a given fuzzy budget and the incompatibility among COTS products. Section 4 makes the conclusions.

2 Elementary Hypotheses

2.1 Primary Symbols

The primary symbols used in this paper are in the following.

m — denotes the number of modules in the giving software system;

n_i — denotes the number of spare COTS in the i^{th} modules, $i = 1, 2, \cdots, m$;

w_i — denotes the weight of the i^{th} modules, $i = 1, 2, \cdots, m$;

q_{ij} — denotes the quality level of the j^{th} spare COTS in the i^{th} modules $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n_i$;

c_{ij} — denotes the cost of the j^{th} spare COTS in the i^{th} modules $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n_i$;

B — denotes the available budget;

$\underset{\sim}{\leq}$ — denotes "approximately less equal";

x_{ij} — denotes the variable in this model;

$$x_{ij} = \begin{cases} 1 & \text{If the } j^{th} \text{ alternative COTS of the } i^{th} \text{ module was chosen} \\ 0 & \text{Otherwise;} \end{cases}$$

2.2 System Prototype

There are four modules in this prototype of the software system ($m = 4$), the architecture of this system prototype is showed in fig.1. From fig.1, we can see that, Program 1 calls upon module 1 and 2, Program 2 calls upon module 2 and 3, Program

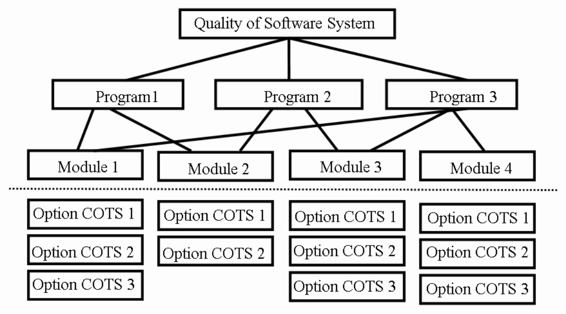

Figure 1: The example of software system

3 calls upon module 1, 3, and 4. The cost of the COTS product is purchasing expense. The quality level can be estimated by the quality model [3] with its value ranged from 0 to 1. The available budget of software system is set to 35. The other data of this system prototype is listed in the following.

$$\begin{array}{llllllll} q_{11}=0.83 & c_{11}=10 & q_{21}=0.85 & c_{21}=8 & q_{31}=0.85 & c_{31}=8 & q_{41}=0.90 & c_{41}=9 \\ q_{12}=0.82 & c_{12}=9 & q_{22}=0.88 & c_{22}=9 & q_{32}=0.79 & c_{32}=7 & q_{42}=0.88 & c_{42}=8 \\ q_{13}=0.78 & c_{13}=8 & & & q_{33}=0.90 & c_{33}=9 & q_{43}=0.81 & c_{43}=8 \end{array}$$

2.3 The Weights of Each Module

At present, there are many methods for setting the weight [4]. According to the access frequency of each module, this paper applies the AHP approach to assign the weight for each module. The details of the computation process are not described for the sake of the length of this paper. The final result is listed in the following.

$$W = (0.416, 0.268, 0.177, 0.140)$$

3 The Fuzzy Optimization Model and Solution

The fuzzy optimization model that this paper has presented is to select the best COTS products for each module subject to available budget and the incompatible limitation among COTS products, to maximize quality of the software system. Since the solving methods of fuzzy optimization models have been presented in many relevant literatures [6], in this paper, we adopt the parameter program method proposed by Verdegay [7].

3.1 The Fuzzy Optimization Model

When making decision for selecting the best COTS product, decision makers are usually faced with the following problems: (1) maximize the system quality subject to a cost constraint, or (2) minimize additional cost under system quality constraints. The fuzzy optimization model in this paper is to select the best COTS products for each module subject to available fuzzy budget, and to maximize quality of the software system based on COTS.

In the development of software system, sometimes a COTS product for one module is incompatible with the alternative COTS products for other modules, due to problems such as implementation technology, interfaces, and licensing. This incompatibility can be denoted by $x_{rs} \leq x_{ut_1}$, that is, if the module r chooses the COTS product s, then module u must choose the COTS product t_1. This condition is called the contingent decision constraint. Suppose that there are two contingent decisions in the model, such as, the alternative COTS$s$for the module r only can be compatible with the COTS products t_1 and t_2 for the module u, that is, if $x_{rs} = 1$, then $x_{ut_1} = 1$ or $x_{ut_2} = 1$, this constraint can be described as $x_{rs} \leq x_{ut_1}$ or $x_{rs} \leq x_{ut_2}$.

Although the above contingent decision does not belong to the linear programming problem, it can be solved by introducing a binary variable y_i. The variable y_i is defined as follows.

$$y_i = \begin{cases} 0 & \text{the } i^{th} \text{ constraint is active} \\ 1 & \text{the } i^{th} \text{ constraint is inactive} \end{cases}$$

In this way, if M is big enough, and only one of constraint condition of the contingent decision of the arbitrary COTS products is allowed to be active, then

$$\begin{aligned} & x_{rs} - x_{ut_i} \leq My_i, \quad i = 1, \ldots, z, \\ & \sum_{i=1}^{z} y_i = z - 1 \\ & y_i = 0/1, \quad i = 1, \ldots, z. \end{aligned}$$

As described above, the fuzzy optimization model is formulated as follows:

$$\max \quad Q = \sum_{i=1}^{m} w_i \left(\sum_{j=1}^{n_i} q_{ij} x_{ij} \right)$$

$$\text{s.t.} \quad \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, \ldots, m, \tag{1}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \underset{\sim}{\leq} B, \tag{2}$$

$$x_{ij} = 0/1, \quad \forall i, j, \tag{3}$$

$$x_{rs} - x_{ut_i} \leq My_i, \quad i = 1, \ldots, z, \tag{4}$$

$$\sum_{i=1}^{z} y_i = z - 1, \tag{5}$$

$$y_i = 0/1, \quad i = 1, \ldots, z. \tag{6}$$

The weighted quality to be maximized has been used as the objective function in software quality studies. In addition, the quality model of the ISO 9126 [3] was developed to assign weights of quality characteristics (attributes) in order to evaluate overall quality of software products. For the constraint (1), one COTS product is selected for each module. This is called the mutually exclusive alternative. In the constraint (2), the available budget is taken as the fuzzy constraint, that is, to allow the constraint to have certain kind to increase and lessen. The symbol, "$\underset{\sim}{\leq}$", denotes "approximately less equal". Therefore, the model has the asymmetrical form of fuzzy linear programming, which can be solved by using fuzzy integer 0/1 programming.

The upper and lower bounds of available budget are defined in the following.

$$\sum_{i=1}^{m} \min_j(c_{ij}) \leq B \leq \sum_{i=1}^{m} \max_j(c_{ij})$$

When B is less than the lower bound, this model is not feasible. If B is greater than the upper bound, then this model has the same optimal solution as the upper bound.

If the goal of system developers is to minimize cost with a given minimum quality level, then this model can be transformed to a cost minimization problem, i.e. by restructuring this model, the objective function becomes a fuzzy constraint and the fuzzy budget constraint (2) becomes the objective function.

3.2 Solving the Model

There are many methods to solve the fuzzy linear programming, the method adopted in this paper is the parameter program method proposed by Verdegay [7]. Its main idea is to consider each α-cut sets in the space of feasible solution constructed by fuzzy constraints, to determine the element set of optimal value, which the objective function is achieved in the α-cut sets. According to the decomposition theorem, the optimal decision sets of the problem can be obtained, only to synthesize all level of the α-cut sets. Then, to use the expansion principle, the optimal objective sets, which are corresponding with the optimal decision sets, further can be obtained.

By using the data of the proposed system prototype in Section 2 and module weights $W = (0.416, 0.268, 0.177, 0.140)$, and adding the following contingent decision constraint:

$$x_{42} \leq x_{11} \text{ or } x_{42} \leq x_{13} \tag{7}$$

The meaning of formula (7) is, if module 4 chooses the second COTS product, module 1 must choose the first COTS product or the third COTS product. This constraint condition can be called as either-or constraint. This model can be expressed in the following.

$$\begin{aligned}
\max \quad & Q = 0.345x_{11} + 0.341x_{12} + 0.325x_{13} + 0.228x_{21} + 0.236x_{22} \\
& \quad + 0.151x_{31} + 0.14x_{32} + 0.16x_{33} + 0.126x_{41} + 0.123x_{42} + 0.113x_{43} \\
\text{s.t.} \quad & x_{42} - x_{11} \leq My_1, \quad x_{42} - x_{13} \leq My_2 \\
& y_1 + y_2 = 1, \quad y_i = 0/1, \quad i = 1,2 \\
& x_{11} + x_{12} + x_{13} = 1,
\end{aligned}$$

Table 1: The Optimal Solutions and The Optimal Objective Values

α	x_{1j}	x_{2j}	x_{3j}	x_{4j}	y_i	Q	α	x_{1j}	x_{2j}	x_{3j}	x_{4j}	y_i	Q
0.0	x_{11}	x_{21}	x_{33}	x_{42}	y_2	0.856	0.6	x_{11}	x_{22}	x_{33}	x_{42}	y_2	0.864
0.1	x_{11}	x_{21}	x_{33}	x_{42}	y_2	0.856	0.7	x_{12}	x_{22}	x_{33}	x_{41}	y_1	0.863
0.2	x_{11}	x_{21}	x_{33}	x_{42}	y_2	0.856	0.8	x_{12}	x_{22}	x_{33}	x_{41}	y_1	0.863
0.3	x_{11}	x_{21}	x_{33}	x_{42}	y_2	0.856	0.9	x_{12}	x_{22}	x_{33}	x_{41}	y_1	0.863
0.4	x_{11}	x_{21}	x_{33}	x_{42}	y_2	0.856	1.0	x_{11}	x_{22}	x_{33}	x_{41}	y_1	0.867
0.5	x_{11}	x_{22}	x_{33}	x_{41}	y_1	0.863							

$$
\begin{aligned}
&x_{21} + x_{22} = 1,\\
&x_{31} + x_{32} + x_{33} = 1,\\
&x_{41} + x_{42} + x_{43} = 1,\\
&10x_{11} + 9x_{12} + 8x_{13} + 8x_{21} + 9x_{22} + 8x_{31} + 7x_{32}\\
&\qquad\qquad + 9x_{33} + 9x_{41} + 8x_{42} + 8x_{43} \underset{\sim}{\leq} 35 \qquad (8)\\
&x_{ij} = 0/1, \quad \forall i; j.
\end{aligned}
$$

According to Verdegay parameter programming method, it is assumed that the biggest deviation of fuzzy constraint, which was allowed by decision maker, is $d = 2$. The formula (8) can be expressed as the following:

$$
\begin{aligned}
&10x_{11} + 9x_{12} + 8x_{13} + 8x_{21} + 9x_{22} + 8x_{31} + 7x_{32} + 9x_{33}\\
&\qquad\qquad + 9x_{41} + 8x_{42} + 8x_{43} \underset{\sim}{\leq} 35 - 2\alpha \qquad (9)
\end{aligned}
$$

For arbitrary given $\alpha \in [0,1]$, the corresponding optimal resolutions may be calculated as $(x_{1j}^{\alpha}, x_{2j}^{\alpha}, x_{3j}^{\alpha}, x_{4j}^{\alpha})$ and the optimal objective values as

$$
\max\ Q^{\alpha} = \sum_{i=1}^{m} w_i \left(\sum_{j=1}^{n_i} q_{ij} x_{ij}^{\alpha}\right).
$$

The details are showed in Tab. 1.

As Tab. 1 shows, with α from 0 to 1, the optimal objective values also change correspondingly, then, the decision maker can select the optimal solution from Tab. 1. If $\alpha = 0$, the optimal solution of this example is $x_{11} = x_{21} = x_{33} = x_{42} = y_2 = 1$, that is, the best selection for modules 1, 2, 3, and 4 is COTS products 2, 2, 3, and 2, respectively, its optimal objective function value is $Q^{\alpha=0} = 0.856$. Since the COTS product quality and module weight have a value between zero and one, the optimal value $Q^{\alpha=0} = 0.856$ means a 85.6% satisfaction level of the software system. Obviously, the fuzzy budget constraint will impact the optimal quality level. In this example, the system quality is changed from $Q^{\alpha=0} = 0.856$ to $Q^{\alpha=1} = 0.867$.

4 Conclusions

In this paper, we have presented a fuzzy optimization model for selecting the best COTS software product in the development of software system based on COTS. A weight to the modules is given by utilizing the Analytic Hierarchy Process (AHP) based on the access frequencies of the modules. The optimization model is subject to available fuzzy budget and contingent decision constraint, which may reflect the real procurement environment of the COTS products. In addition, our model assumes that there is no cost in developing interface programs to connect between modules, and the operation of COTS product is statistically independent. Therefore, in actual application, the assumptions of the model should be noticed.

References

[1] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*, Fifth Edition, McGraw-Hill Companies, 2001.

[2] B. Craig Meyers and Patricia Oberndorf. *Managing Software Acquisition Open Systems and COTS Products*, Addison Wesley, 2002.

[3] *ISO/IEC 9126, Information Technology - Software Product Evaluation — Quality Characteristics and Guidelines for Their Use*, International standard, ISO, Geneve, Switzerland, 1991.

[4] Chaoyuan Yue. *Decision Making Theory and Methods*, Science Press, Beijing, 2003.

[5] F. Zahedi and N. Ashrafi. Software reliability allocation based on structure, utility, price, and cost. *IEEE Trans. on Software Engrg.*, 17(4), 345–356, 1991.

[6] Jiupin Xu and Jun Li. *Multiple Objectives Decision Making Theory and Methods*, Tsinghua University Press, Beijing, 2005.

[7] J. L. Verdegay. Fuzzy mathematical programming. M. M. Gupta, E. Sanchez eds., *Fuzzy Information and Decision Process*, 231–237, 1982.