Fuzzy Approach for Selecting Optimal COTS Based Software Products Under Consensus Recovery Block Scheme

P. C. Jha[1], Shivani Bali[2] and **P. K. Kapur[3]**

Submitted in March 2010; Accepted in December 2010

Abstract - The cost associated with development of a large and complex software system is formidable. In today's customer driven market, improvement of quality aspects in terms of reliability of the product is also gaining increased importance. But the resources are limited and the manager has to maneuver within a tight schedule. In order to meet these challenges, many organizations are making use of Commercial Off-The-Shelf (COTS) software. This paper develops a fuzzy multi objective optimization model approach for selecting the optimal COTS software product among alternatives for each module in the development of modular software system. The problem is formulated for consensus recovery block fault tolerant scheme. In today's ever changing environment, it is arduous to estimate the precise cost and reliability of software. Therefore, we develop a fuzzy multi objective optimization models for selecting optimal COTS software products. Numerical illustrations are provided to demonstrate the models developed.

Index Terms - Modular software, software reliability, COTS products, fault tolerance, fuzzy optimization.

1.0 INTRODUCTION

In our modern society, computers are used in diverse areas for various applications, for example, air traffic control, nuclear reactors, aircraft, real time military, industrial process control, and hospital patient monitoring systems. As the functionality of computer operations becomes more essential and complicated and critical software operations becomes more essential and complicated and critical software applications increase in size and complexity, there is a greater need for computer software reliability.

Software reliability is an important attribute of software quality, together with functionality, usability, performance, serviceability, capability, install ability, maintainability, and documentation. Software reliability is hard to achieve, because the complexity of software tends to be high. While any system with a high degree of complexity, including software, will be hard to reach a certain level of reliability, system developers tend to push complexity into the software layer, with the rapid growth of system size and case of doing so by upgrading the software.

[1,3]Department of Operational Research, University of Delhi, Delhi, INDIA
[2]Lal Bahadur Shastri Institute of Management, Delhi, INDIA
E-Mail: [1]lbsshivani@gmail.com

Commercial off-the-shelf (COTS) components engineering is an emerging paradigm for software development. Benefits of COTS based development include significant reduction in the development cost, time and improvement in the dependability requirement. Commercial off-the-shelf (COTS) components are used without any code modification and inspection. The components, which are not available in the market or cannot be purchased economically, can be developed within the organization. Component Based Software Engineering (CBSE) process model has become a kind of process model of software development project [6,9] Respective developers of the components provide information about their quality normally in terms of reliability. COTS components are received from distributor and are used 'as is'. No changes are normally made to their source codes. Only the code that is necessary to integrate these products is required to be developed in house. Large software systems have modular structures. The advancement of technology has made the use of COTS products as modules a possibility. A component can now be chosen for a module from the number of alternatives available in the market.

This paper proposes fuzzy multi objective optimization models for selecting the best COTS software product for each module. Software whose failure can have severe repercussions can be made fault tolerant through redundancy at module level [1]. Because of our present inability to produce error-free software, software fault tolerance is and will continue to be an important consideration in software systems. For some applications software safety is important and fault tolerance techniques used in those applications are aimed at preventing catastrophes. Multi version software fault tolerance techniques are based on the assumption that software built differently should fail differently and thus, if one of the redundant version fails, at least one of the others should provide an acceptable output. In [3, 4] reliability optimization problems for fault tolerant systems have been discussed. The authors have discussed two reliability models. In this paper, a fault tolerance architecture, which support consensus recovery block Scheme is proposed.

In the existing research in this area it is assumed that a crisp or a constant value of all the parameters is known. Jha et al formulated bi-criteria optimization model for selection of COTS based software system for consensus recovery block scheme by taking crisp estimates of reliability and cost [5]. However, in practice, it is not possible for a management to get precise value of reliability and cost for a software system. Or it may happen that they decide not to set precise levels due to the market considerations and are ready to have some tolerance of their objectives. When the precise values of parameter of the problem are not known, the problem becomes a fuzzy optimization problem and the solution so obtained is a fuzzy approximation. Gupta et. al proposed a hybrid approach for selecting the optimal COTS software product in the development of modular software system[8].

This paper proposes two fuzzy multi-objective optimization models for selecting the best COTS software product for each module. The first optimization model (optimization model-I) of this paper is a joint optimization problem that maximizes the system reliability with simultaneously minimizing cost. The second optimization model (optimization model-II) considers the issue of compatibility between different alternatives of modules as it is observed that some COTS components cannot integrate with all the alternatives of another module. We assume the existence of virtual versions, apart from available versions, having negligible reliabilities and zero costs. Virtual versions are chosen only when we have insufficient budget. In a situation where this particular version is chosen, the corresponding alternative is not to be added to the system. The rest of the paper is organized as follows. Section 2 proposes notations. In section 3, we develop a crisp model and describe non –linear S-shape fuzzy membership functions in respect of both the chosen objectives, viz. the reliability and the cost. In this section, we also present fuzzy multi-objective optimization models for selecting the best COTS product for each module. Section 4 paper are illustrated with numerical example. Section 5, we furnish our concluding observations.

2.0 NOTATIONS

R : System quality measure

f_l : Frequency of use, of function l

s_l : Set of modules required for function l

R_i : Reliability of module i

L : Number of functions, the software is required to perform

n : Number of modules in the software.

m_i : Number of alternatives available for module i

V_{ij} : Number of versions available for alternative j of module i

c_{ijk} : Cost of version k of alternative j of module i (COTS)

t_1 : Probability that next alternative is not invoked upon failure of the current alternative

t_2 : Probability that the correct result is judged wrong.

t_3 : Probability that an incorrect result is accepted as correct.

Y_{ij} : Event that correct result of alternative j of module i is accepted.

X_{ij} : Event that output of alternative j of module i is rejected.

r_{ij} : Reliability of alternative j of module i

r_{ijk} : Reliability of version k of alternative j of module i

z_{ij} : Binary variable taking value 0 or 1

$$\begin{cases} 1, & \text{if alternative } j \text{ is present in module } i\text{ s} \\ 0, & \text{otherwise} \end{cases}$$

3.0 MULTI-OBJECTIVE OPTIMIZATION MODELS SELECTING COTS PRODUCTS

In this section, we formulate COTS software products selection problem as an optimization problem with multiple objectives. The first optimization model is developed for the following situations, which also holds good for the second model, but with additional assumptions related to compatibility among alternatives of a module.

The following are the assumptions of optimization Models:

3.0.1 There is a specified budget for the development of software system.

3.0.2 A software system consists of a finite number of modules.

3.0.3 A software system is required to perform a known number of functions. The program written for a function can call a series of modules $(\leq n)$. A failure occurs if a module fails to carry out an intended operation.

3.0.4 Codes written for integration of modules don't contain any bug.

3.0.5 Several alternatives are available for each module. Fault tolerant architecture is desired in the modules (it has to be within the specified budget). Independently developed alternatives (primarily COTS components) are attached in the modules and work similar to the recovery block scheme discussed in [3,4].

3.0.6 The cost of an alternative is the development cost, if developed in house; otherwise it is the buying price for the COTS product. Reliability for all the components are known and no separate testing is done.

3.0.7 Different versions with respect to cost and reliability of a module are available.

3.0.8 Other than available cost-reliability versions of an alternative, we assume the existence of a virtual versions, which has a negligible reliability of 0.001 and zero cost. These components are denoted by index one in the third subscript of x_{ijk}, c_{ijk} and r_{ijk}. for example r_{ij1} denotes the reliability of first version of alternatives j for module i, having the above property.

3.1 Multi-Objective Optimization Model I

In the first optimization model it is assumed that the alternatives of a module are in a consensus recovery block [10]. Consensus recovery block requires independent development of independent alternatives of a program, which the COTS components satisfy and a voting procedure. Upon invocation of the consensus recovery block all alternatives are executed and their outputs are submitted to a voting procedure. Since it is assumed that there is no common fault, if two or more alternatives agree on one output then that output is designated as correct. Otherwise the next stage is entered. At this stage the best version is examined by an acceptance test. If the output is accepted, it is treated as the correct one. However if the output is not accepted, the next best version is subject to testing. This process continues until an acceptable output is found or all outputs are exhausted.

Problem (P1)

Maximize $R = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i$ (1)

Minimize $C = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} c_{ijk} x_{ijk}$ (2)

Subject to

$X \in S = \{ X_{ijk}$ is binary variable

$$R_i = 1 + \left[\sum_{j=1}^{m_i} \frac{1}{(1-r_{ik})^{z_{ik}}} \left[\prod_{k=1}^{m_i} (1-r_{ik})^{z_{ik}} \right] \left[1 - \left(1-r_{ij}\right)^{z_{ij}} \right] + \prod_{j=1}^{m_i} \left(1-r_{ij}\right)^{z_{ij}} \right] \quad (3)$$

$$\left[\sum_{j=1}^{m_i} z_{ij} \left[\prod_{k=1}^{j-1} P\left(X_{ik}\right)^{z_{ij}} \right] P\left(Y_{ij}\right)^{z_{ij}} - 1 \right] \; ; \; i = 1, 2, \dots\dots\dots n$$

$$P(X_{ij}) = (1-t_1)\left[(1-r_{ij})(1-t_3) + r_{ij} t_2\right]$$

$$P(Y_{ij}) = r_{ij}(1-t_2)$$

$$r_{ij} = \sum_{k=1}^{V_{ij}} x_{ijk} r_{ijk} \quad j = 1, 2, \dots\dots, m_i \text{ and } i = 1, 2, \dots\dots n \quad (4)$$

$$\sum_{k=1}^{V_{ij}} x_{ijk} = 1, \text{ for } j = 1, 2, \dots\dots, m_i \text{ and } i = 1, 2, \dots\dots, n \quad (5)$$

$$x_{ij1} + z_{ij} = 1; \; j = 1, 2, \dots\dots m_i \quad (6)$$

$$\sum_{j=1}^{m_i} z_{ij} \geq 1 \, ; \, i = 1, 2, \dots\dots, n \quad (7)\}$$

Objective function (1) maximizes the system quality (in terms of reliability) through a weighted function of module reliabilities. Reliability of modules that are invoked more frequently during use is given higher weights. Analytic Hierarchy Process (AHP) can be effectively used to calculate these weights and (2) minimize the overall cost of the system.

Constraint (3) estimates the reliability of module i. As it has been assumed that the exception raising and control transfer programs work perfectly, a module fails if all attached alternatives fail.

Constraint (5) ensures that exactly one version is chosen from each alternative of a module. It includes the possibility of choosing a dummy version. Equation (6) and (7) guarantee that not all chosen alternatives of module are dummies. Optimization model-I is a 0-1 Bi-Criterion integer programming problem. An example is solved using software package LINGO.

It is observed that some alternatives of a module may not be compatible with alternatives of another module. The next optimization model II addresses this problem. It is done, incorporating additional constraints in the optimization models. This constraint can be represented as $x_{gsq} \leq x_{hu_t c}$, which means that if alternative s for module g is chosen, then alternative $u_t, t = 1, \dots\dots\dots z$ have to be chosen for module h. We also assume that if two alternatives are compatible, then their versions are also compatible.

$$x_{gsq} - x_{hu_t c} \leq M y_t$$

$$q = 2, \dots\dots, V_{gs}, \; c = 2, \dots\dots, V_{hu_t}, \; s = 1, \dots\dots, m_g \quad (8)$$

$$\sum y_t = z\left(V_{hu_t} - 2\right) \quad (9)$$

Constraint (9) ensures that only one alternative is compatible.

Constraint (3) to (7) is equivalent to problem (P1). Constraint (8) and (9) make use of binary variable y_t to choose one pair of alternatives from among different alternative pairs of modules. If more than one alternative compatible component is to be chosen for redundancy, constraint (9) can be relaxed as follows.

$$\sum y_t \leq z\left(V_{hu_t} - 2\right) \quad (10)$$

Constraint (10) ensure more than one alternative is compatible.

3.2 Multi-Objective Optimization Model II

Problem (P1) can be transformed to another optimization problem using compatibility constraint as follows.

Maximize $R = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i$

Minimize $C = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} c_{ijk} x_{ijk}$

Subject to

$$X \in S$$

$$x_{gsq} - x_{hu_t c} \leq M y_t$$

$$q = 2, \dots\dots, V_{gs}, \; c = 2, \dots\dots, V_{hu_t}, \; s = 1, \dots\dots, m_g$$

$$\sum y_t = z\left(V_{hu_t} - 2\right)$$

$$\sum y_t \leq z\left(V_{hu_t} - 2\right)$$

Similar constraints can be written for all pairs of compatible modules.

3.3 Selection Model For Cots Software Products Based On Fuzzy Decision Theory

The model formulation for the above said problem requires an estimate of reliability and cost for various alternative COTS in the modules. Due to the changing environment, these estimates cannot be determined definitely because cost and reliability are affected by ambiguous and uncertain factors which cannot be measured precisely. Also the decision maker's assessment about these estimates may be based on incomplete knowledge about the COTS product itself and other aspects (e.g. vendor's credentials).Under such conditions; making a decision based upon crisp model is not the best decision. Since software development cost is ever changing and it becomes difficult to estimate the definite cost and reliability of the software. Therefore the issue of selecting COTS software products becomes the one of a choice from a fuzzy set of subjective/intuitive interpretations, the term fuzzy being suggestive of the diversity of both the decision maker's objective functions as well as that of the constraints.

Therefore, we formulate fuzzy multi-objective optimization model for COTS software products selection based on vague aspiration levels, the decision maker may decide his aspiration levels on the basis of past experience and knowledge possessed by him. To express vague aspiration levels of the decision, various membership functions have been proposed [13, 14]. A fuzzy linear programming problem with non linear membership function results in a non linear programming problem. Usually, a linear membership function is employed to avoid nonlinearity. Also, if membership function is interpreted as the fuzzy utility of the decision maker, which describes the behavior of indifference, preference or aversion towards uncertainty, a non linear membership function is a better representation than a linear membership function.

In this paper, we use a logistic function [12], i.e. a non linear S-shape membership function to express vague aspiration levels of the decision maker. The S-shape membership function is given by

$$f(x) = \frac{1}{1+\exp(-\alpha x)}$$

where $\alpha,\ 0 < \alpha < \infty$ is a fuzzy parameter which measures the degree of vagueness. The reason why we use this function is that, it is easily handled. Also, the logistic membership function preserves linearity even when the operator "product" is used instead of the operator "min" to aggregate the overall satisfaction to arrive at the fuzzy set decision.

In the MOP model proposed in Section 3.1 and 3.2, the two objectives i.e. the reliability and the cost are considered to be ambiguous and uncertain. We use the following nonlinear S-shape membership functions to express the vague aspiration levels.

- The membership function of the goal for the reliability is given by

$$\mu_R(x) = \frac{1}{1+\exp\left(-\alpha_R\left(\sum_{l=1}^{L} f_l \prod_{i\in s_l} R_i - \mathrm{R}_m\right)\right)}$$

where R_m is the mid-point (middle aspiration level for the reliability) at which the membership function value is 0.5 and α_R can be given by decision maker based on his own degree of satisfaction.

- The membership function of the goal for the cost is given by

$$\mu_C(x) = \frac{1}{1+\exp\left(\alpha_C\left(\sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=1}^{V_{ij}} c_{ijk}x_{ijk} - C_m\right)\right)}$$

where C_m is the mid-point (middle aspiration level for the cost) at which the membership function value is 0.5 and α_C can be given by decision maker based on his own degree of satisfaction.

Following Bellman-Zadeh's Maximization principle [2] and using the above defined fuzzy membership functions, the fuzzy multi-objective optimization model for selecting the COTS software products is formulated as follows:

Problem P

$$\max\ \lambda$$

$$s.t\ \lambda \le \mu_R(x),$$

$$\lambda \le \mu_C(x),$$

$$0 \le \lambda \le 1,$$

and the constraints (3) to (7).

Fuzzy multi-objective optimization model (P) is solved for maximized degree of membership for the fuzzy decision. In this approach all the fuzzy objectives are treated equivalently. However, approaches have been discussed in literature with situations in which the objectives are not equally important [7, 11].

4.0 ILLUSTRATIVE EXAMPLES

Consider a software system having two modules with more than one alternative for each module. The cost reliability data set is given in Table-1. Note that the cost of first version i.e. the virtual versions for all alternatives is zero and reliability is 0.001. This is done for the following reason: If in the optimal solution, for some module $x_{ij1} = 1$, that implies corresponding alternative is not to be attached in the module.

Let $L = 3$, $s_1 = \{1,2\}$, $s_2 = \{1\}$, $s_3 = \{2\}$, $f_1 = 0.5$, $f_2 = 0.3$ and $f_3 = 0.2$.

It is also assumed that $t_1 = .01, t_2 = .05$ and $t_3 = .01$

Structure of Software

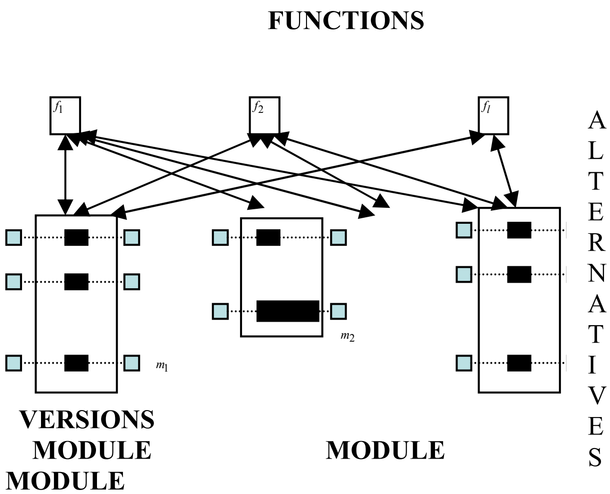

Figure 1: Structure of the software

Cost and Reliability Dataset

Modules	Alternatives	Versions 1		2		3	
		Cost	Reliability	Cost	Reliability	Cost	Reliability
1	1	0	0.001	8.2	.90	9.0	.88
	2	0	0.001	7.5	.86	9.0	.92
	3	0	0.001	8.5	.90	9.5	.88
2	1	0	0.001	3.2	.87	4.0	.86
	2	0	0.001	3.4	.91	4.3	.89
	3	0	0.001	5.0	.89	6.8	.86
	4	0	0.001	4.8	.86	6.8	.88

By taking $\alpha_R = 0.60$ and $\alpha_c = 16$

4.1 Optimization Model I

The problem is solved using software package LINGO [8]. Following solution is obtained.

$$x_{111} = x_{122} = x_{132} = 1$$

$$x_{211} = x_{222} = x_{232} = x_{242} = 1$$

It is observed that two or more alternatives are chosen for each module. Redundancy is allowed for both the modules. The system reliability for the above solution is 0.79 and cost is 30.5 units and the achievement level of membership function is $\lambda = 0.58$.

4.2 Optimization Model- II

To illustrate optimization model for compatibility, we use previous results. We assume second alternative of second module is compatible with second and third alternatives of first module. Following solution was obtained using LINGO.

$$x_{111} = x_{123} = x_{132} = 1$$

$$x_{211} = x_{222} = x_{223} = x_{243} = 1$$

It is observed that due to the compatibility condition, second alternative of first module is chosen as it is compatible with second alternative of second module. The system reliability for the above solution is 0.79 and cost is 32 units and the achievement level of membership function is $\lambda = 0.58$.

5.0 CONCLUSION

In this paper, fuzzy multi-objective optimization model approach for selecting the optimal COTS software product among alternatives for each module in the development of modular software system is discussed. The problem is formulated for consensus recovery block fault tolerant scheme. In today's ever changing environment, it is arduous to estimate the precise cost and reliability of software. For such situation where the software is developed by assembling COTS software products, then it is not possible to get the crisp estimates of cost and reliability of these COTS products. Therefore, we have drawn on fuzzy methodology for the estimation of reliability and cost. This developed approach can effectively deal with the vagueness and subjectivity of expert information.

REFERENCES

[1]. Belli F, Jadrzejowich P. "An approach to reliability optimization of software with redundancy" IEEE transactions on Software Engineering 1991; 17(3): 310-312

[2]. Bellman, R., Zadeh, L.A.:"Decision making in a fuzzy enviornment", Management Science 17, 1970; 141-164

[3]. Berman O, Dinesh Kumar U. "Optimization models for recovery block schemes" European Journal of Operational Research 1999; 115: 368-379

[4]. Dinesh Kumar U, "Reliability Analysis of fault tolerant recovery block", OPSEARCH, 1998; 35(2), 281-294

[5]. Jha P.C. , Shivani Vaid and Kapur P.K. " Optimal Component Selection For Fault Tolerance COTS Based Software Under Consensus Recovery Block Scheme", INDIACoM-2009,387-390

[6]. Meyers, B.C., Oberndorf, P., "Managing Software Acquisition Open systems and COTS products." Addison-Wesley, Reading (2002)

[7]. Narasimhan, R.: "On fuzzy goal programming – some comments",Decision Sciences 12;1981; 532-538

[8]. Gupta P., Mehlawat M.K., Mittal G., Verma S,"A Hybrid Approach for selecting Optimal COTS Products ", Computional Science and its Application-ICCSA 2009;Springer publication; 5592; 949-962.

[9]. Roger, S.P.: "Software Engineering: A Practitoner's Approach", McGraw Hill Companies, 5th Edition, New York 2001
[10]. Scott RK, Gautt JW and Mc Alliter DF "Fault tolerant software reliability modelling" IEEE transactions on Software Engineering 1987; 582-592
[11]. Tiwari, R.N., Dharmar, S., Rao, J.R.: "Fuzzy goal programming – an additive model", Fuzzy Sets and systems24, 1987;27-34
[12]. Watada, J.:" Fuzzy portfolio selection and its applications to decision making",Tatra Mountains Mathematical Publications 13, 1997; 219-248.
[13]. Zimmermann, H.-J.:"Description and optimization of fuzzy systems". International Journal of General Systems 2; 1976;209-215
[14]. Zimmermann, H.-J.:"Fuzzy programming and linear programming with several objective functions." Fuzzy Sets and Systems 1; 1978; 45-55

Continued from page no. 317

[18]. Routray S., Sherry A. M., Reddy B. V. R.(2005), "A Genetic Algorithm Approach for Dynamic Routing of ATM Networks", *Paradigm* Vol. IX, No. 1, January-June 2005, pp 86-92.
[19]. Routray S., Sherry A. M., Reddy B. V. R.(2006), "Bandwidth Optimization through Dynamic Routing in ATM Networks: Genetic Algorithm &Tabu Search Approach" IJCS *International Journal of Computer Science*, Vol. 1., No. 3, pp 188-194.
[20]. Routray S., Sherry A. M., Reddy B. V. R.(2006), "Handoff Scheme for Wireless ATM Network: A Genetic Algorithm Approach" *The 4th International Conference on Computing, Communications and Control Technologies*: CCCT '06, 2006.
[21]. S. Routray, A. M. Sherry, B. V. R. Reddy, "ATM network planning: A genetic Algorithm Approach", Journal of Applied & Theoretical Information Technology (JATIT), available through EBSCO Publishing USA, Volume 3 No.4, December 2007 pp 72-79.
[22]. Srinivas M., Patnaik Lalit M.(1994), "Genetic Algorithms: A survey", *IEEE*, 1994, 17-26.
[23]. Srinivas M., Patnaik Lalit M. (1994), "Adaptive probabilities of crossover and Mutation in Genetic Algorithms", *IEEE Trans. System*,1994, 656-667.
[24]. Thompson D. R., Bilbro G. L.(2000), "Comparison of a genetic algorithm with a simulated annealing algorithm for the design of an ATM network", *Communications Letters, IEEE* Volume 4, Issue 8, Page(s):267 – 269.
[25]. Wong P. and Britland D.(1993), " Mobile Data Communication ", Artech House, 1993.
[26]. Wu T., Kolar D. J., and Cardwell R. H.(1998), "Survivable Network Architectures for Broadband Fiber Optic Networks: Models and Performance Comparisons", *IEEE Journal of Lightwave Technology*, vol. 6, no. 11, pp. 1698-1709.