Optimal Component Selection Problem for Cots Based Software System under Consensus Recovery Block Scheme: A Goal Programming Approach

Deepak Kumar
AIIT
Amity University India

P. C. Jha
Department of Operational Research, University of Delhi, India

P. K. Kapur
Department of Operational Research, University of Delhi, India

U. Dinesh Kumar
Indian Institute of Management, Bangalore, India

ABSTRACT
Cost, reliability and time are the three main quality attributes of a software system. Now days much software are designed on COTS component in order to facilitate timely development with reduced cost and improved reliability. Software deigned to handle critical control systems have very high reliability requirements. Fault tolerance is designed in these systems for some or all of the software modules so that execution can be resumed even after failure with minimal loss of data and time. Designing fault tolerance requires extra resources. Even though reliability requirement are very high the developers can't spend endless resources on any project. This is a trade off problem between reliability and cost. Many such problems have been discussed in literature considering distinct objectives and constraints and have given good results. An effective approach to discuss this problem is to formulate a multi-objective problem with cost minimization and reliability maximization as the two objectives with an upper bound on cost and lower bound on reliability. In this paper we formulate this bi-criteria problem and discuss the solution methodology. The problem is formulated for consensus recovery block fault tolerant scheme. In case a feasible solution for the problem exists, criterion vector approach is used to solve the problem and otherwise if the bounds are contradictory a goal programming approach is used to solve the problem to obtain a compromised solution. Alternative goal solutions are obtained assigning different weights for the objective to facilitate the decision maker with correct decision.

Keywords
Software reliability, fault tolerance, COTS products, optimization, goal programming, trade off problem.

1. INTRODUCTION
Computer and its related technology invade every aspect of life. Economical, reliable, and quality software are necessary for any organization for proper functioning. Software should not contain any errors when released to the users. Traditionally the quality of software is expressed in terms of reliability, reliability being the main software quality characteristic. Software reliability measures how well a software system operates to meet the user requirement. Software systems which are developed to operate and control the functioning of some sophisticated and critical systems must be developed in such a way that their execution can be resumed even on a failure in the system with minimal loss of data and time. Such software systems which can continue execution even on a failure are called fault tolerant software. Fault tolerant software has been considered for use in a number of critical application areas like nuclear power plant, military, emergency services, air traffic control etc. Fault Tolerance is the property that enables a system (often computer-based) to continue operating properly in the event of the failure of (or one or more faults within) some of its components. There are two basic structural methodologies for fault tolerant system namely--- Recovery Block scheme and N-Version Scheme. The basic mechanism of both the schemes is to provide redundant software to tolerate software failures. There are some other Fault Tolerant System that combines both Recovery Block schemes and N-Version Schemes to create new hybrid system, such as Consensus Recovery Block. In this system if N-Version programming fails, the system reverts to Recovery Block using the same modules. Only when both N-Version programming and Recovery Block fails does the system fails. A careful use of redundancy may allow the system to tolerate faults generated during software design and coding thus improving software reliability.

The redundant components in software for building the fault tolerance must be build following the design diversity techniques to minimize the possibilities of common errors. Software developers use mostly COTS (Commercial off-the-shelf) components for the redundant software modules. COTS components are used as alternatives to in-house developments. Doing so developers can make significant savings in procurement and maintenance and sometimes they are more reliable. Together with this developers of the commercial product integrate new technologies and new standard into the product faster than in house development. However COTS software specifications are written by external sources, organization are sometimes wary of these products because they fear that future changes to the product will not be under their control.

Improving software reliability, using redundancy, however, requires additional resources, i.e. additional cost in terms of redundant software and hardware requirement. Every developer has some constraint on budget how high reliability requirements may be. Therefore the redundancy level to achieve fault tolerance must be carefully determined, and if possible, optimized. Many such problems have been discussed in literature considering distinct objectives and constraints and have given good results. The main consideration in these problems is either cost minimization or reliability maximization. Achieving the highest possible level of reliability is the primary concern of the developers of fault tolerant software but minimizing the development cost can't be ignored. If the cost minimization is overlooked the

optimization routine may select all of the components with highest possible reliability considerably increasing the cost. Hence an effective approach to discuss this problem is to formulate this optimization problem as a multi-objective problem with cost minimization and reliability maximization as the two objectives with an upper bound on cost and lower bound on reliability. In this paper we formulate this bi-criteria problem and discuss the solution methodology. The problem is formulated for consensus recovery block fault tolerant scheme. In case a feasible solution for the problem exists, criterion vector approach is used to solve the problem and otherwise if the bounds are contradictory a goal programming approach is used to solve the problem to obtain a compromised solution. Alternative goal solutions are obtained assigning different weights for the objective to facilitate the decision maker with correct decision.

This paper is classified as follows: Section 2 describes the problem specification, assumptions and notations of the optimization model. In the section 2.1 the optimization model of the problem is formulated and the solution methodology is illustrated. Section 3 numerical illustrations are provided to solve the problem. Finally conclusions are drawn in section 4.

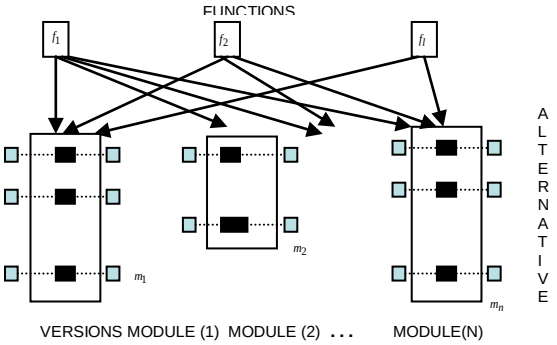

Fig. 1: Software Structure of the Software

Software modules which are called very frequently or whose function is critical to the software operation can be made fault tolerant using redundant modules implemented using some fault tolerant scheme. It may be true for all or none of the modules [2]. Now in the optimization problem under consideration we would like to determine the optimal constitution of the software so that reliability is maximized and cost is minimized. The optimization problem determines the level of redundancy for each module level and the version of the alternative to choose at that redundancy level. For the problem formulation we also consider the fact not all the functions have same level of importance hence they can be assigned weights according to their relative importance, which can be determined on the basic of the frequency of usage. We also 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. Apart from these specifications the other considerations and assumptions for the problem formulation are as follows

Assumptions

1. There budget for the limited and known.
2. 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. Codes written for integration of modules don't contain any bug.
4. Redundant modules are implemented on the software using Consensus Recovery Block scheme [3, 4].
5. The cost of a version of 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.
6. 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.

2. PROBLEM FORMULATION

Here we consider a software system developed on a set of modules (say n). The software is developed to perform a set of functions (say l) using the modules. Different functions call different set of modules for its functioning. Some function may call all while others may call only one of the software modules. Modules of the software are COTS products. Different alternatives of all the modules are available in the market at different cost and level of reliability. Furthermore for each alternative of modules different versions available which also differ in terms of cost and reliability. A schematic representation of the software system can be given by figure 1.

Notation

R : System reliability measure

L :

Number of functions, the software is required to perform

f_l : Frequency of use, of function l

s_l : Set of modules required for function l

R_i : Reliability of module i

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 (Including virtual version)

C_{ijk}:

Cost of version k of alternative j for module i

t_1 :

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

t_2 :

Probability that a 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

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

r_{ijk}:

Reliability of version k of alternative j for module i

K : Total budget available for all the modules

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

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

$$X_{ijk} = \begin{cases} 1, & \text{if version } k \text{ of alternative } j \text{ is present in module } i \\ 0, & \text{otherwise} \end{cases}$$

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

R^l : Reliability of l^{th} function

2.1 Optimization Model

In this optimization model, our aim is to maximize reliability and minimize total cost of the software system. It is assumed that various version of different alternatives of a module are in a Consensus Recovery Block. Upon invocation of a module first alternative is executed and the result is submitted for acceptance test. If it is rejected, the second alternative is executed with the original inputs. The same process continues through all the alternatives until a result is accepted or the Consensus Recovery Block (module) fails. By introducing redundancies fault tolerance is achieved and increasing the number of redundancies the possibility that the Consensus recovery block terminates with a failure is reduced, i.e. reliability is increased.

Objective Functions

The reliability of function l is defined as $R^l = \prod_{i \in S_l} R_i$

Then system reliability is

Maximize $R = \sum_{l=1}^{L} f_l R^l$

Where

$$\sum_{l=1}^{L} f_l = 1; \quad f_l \geq 0, \ l = 1, 2, \ldots\ldots, L$$

Hence the reliability objective function is

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

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

(1)

Total cost is the sum of cost of selected version of different alternative of each module. Hence the cost objective is

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

The management always want to achieve a certain level of reliability (Ro) under limited resources (C0). Therefore optimization problem considering all the specifications and assumptions is stated as

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

Minimize

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

Subject to

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

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

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

$$r_{ij} = \sum_{k=1}^{V_{ij}} x_{ijk} r_{ijk}$$

$j = 1, 2, \ldots\ldots, m_i$ and $i = 1, 2, \ldots\ldots n$

$$\sum_{k=1}^{V_{ij}} x_{ijk} = 1, \ \text{for } j = 1, 2, \ldots\ldots, m_i \ \text{and } i = 1, 2, \ldots\ldots, n$$

(5)

$$x_{ij1} + z_{ij} = 1; \ \ j = 1, 2, \ldots\ldots m_i$$

$$\sum_{j=1}^{m_i} z_{ij} \geq 1 \ ; \ i = 1, 2, \ldots\ldots, n$$

}

$$\sum_{l=1}^{L} f_1 \prod_{i \in S_i} R_i \geq R_o \quad (8)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} C_{ijk}\ x_{ijk} \leq C \quad (9)$$

where X is a vector consisting of x_{ijk}; $i = 1,...,n;\ j = 1,...,m_i;\quad k = 1,...,V_{ij}$

In the above problem (P1), objective function (1) maximizes the reliability of software system through a weighted function of module reliabilities. Reliability of modules that are invoked more frequently during use are given higher weights. Analytic Hierarchy Process (AHP) can be effectively used to calculate these weights and (2) minimize the total cost of the software.

Equation (3) estimates the reliability of module i following the Consensus Recovery Scheme. 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. Equations (6) and (7) guarantees that not all chosen alternatives of modules are dummies. (8) specifies the upper bound on the reliability and (9) is the budget constraint. Optimization model (P1) is a 0-1 Bi-Criterion integer programming problem. An example is solved using software package LINGO after normalization.

Normalization

The problem (P1) discussed above is Bi- criteria optimization problem in which on one hand system reliability is maximized and other hand total cost is minimized. Reliability is measured as probability having values between [0, 1], whereas cost is measured in different unit. Before the problem can be solved it needs to be normalized by expressing both objectives in the same units. For this purpose we use the following transformation to express the cost objective having values between [0, 1].

$$\bar{C}_{ijk} = C_{ijk} / C$$

The cost objective in the problem (P1) then can be rewritten as

$$\text{Minimize} \qquad C(X) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk}$$

and the budget constraint is reformulated as

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk} \leq 1$$

The problem (P2) can further be written as vector optimization problem as follow

Vector Max $F(X)$

Subject to

$$X \in S$$

$$\sum_{l=1}^{L} f_1 \prod_{i \in S_i} R_i \geq R_0 \quad (8)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk} \leq 1 \quad (9)$$

(P2)

where $F(X) = (F_1 = R,\ F_2 = -C(X))^T$

Finding Properly Efficient Solution

Definition 1 [7]: A feasible solution $X^* \in S$ is said to be an efficient solution for the below problem if there exists no $X \in S$ such that $F(X) \geq F(X^*)$ and $F(X) \neq F(X^*)$

Definition 2 [7]: An efficient solution $X^* \in S$ is said to be an properly efficient solution for the problem (P2) if there exist $\alpha > 0$ such that for each r $(F_r(X) - F_r(X^*)) / (F_j(X^*) - F_j(X)) < \alpha$ for some j with $F_j(X) \leq F_j(X^*)$ and $F_r(X) > F_r(X^*)$ for $X \in S$.

Above optimization problem can be rewritten by using Geofrion Scalarization [4]. Objective function of problem (P3) can be rewritten by considering weighting vectors (λ_1, λ_2) as follow

Maximize $\lambda_1 F_1 + \lambda_2 F_2$

Subject to

$$X \in S$$

$$\sum_{l=1}^{L} f_1 \prod_{i \in S_i} R_i \geq R_0$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk} \leq 1$$

(P3)

$$\lambda_1 + \lambda_2 = 1 \qquad \lambda_1, \lambda_2 \geq 0$$

Lemma 1 [4]: The optimal solution of the problem (P3) for fixed λ_1 and λ_2 is a properly efficient solution for the problem (P2) and consequently (P1).

Hence the final formulation of the problem is

$$\text{Maximize } \lambda_1 \left[\sum_{l=1}^{L} f_l \prod_{i \in sl} R_i \right] - \lambda_2 \left[\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk} \right]$$

Subject to

$$X \in S$$

$$\sum_{l=1}^{L} f_1 \prod_{i \in S_i} R_i \geq R_0 \quad (8)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \bar{C}_{ijk}\ x_{ijk} \leq 1 \quad (9)$$

(P4)

$\lambda_1 + \lambda_2 = 1 \quad \lambda_1, \lambda_2 \geq 0$

If the problem (P4) is not feasible for any fixed value of λ_1 and λ_2 (relative importance) specified by the management then a compromised solution to the problem can be obtained using goal programming approach. The Goal programming problem for the problem (P3) is formulated as follows

Minimize $\lambda_1 \eta_1 - \lambda_2 \rho_2$

Subject to

(P5)

$X \in S$

$$\sum_{l=1}^{L} f_l \prod_{i \in S_l} R_i + \eta_1 - \rho_2 = R_o$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \overline{C}_{ijk}\, x_{ijk} + \eta_1 - \rho_2 = \frac{C_0}{\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} C_{ijk}}$$

$\lambda_1 + \lambda_2 = 1 \quad \lambda_1, \lambda_2 \geq 0$

In the next section we provide a numerical illustration for the problem.

3. ILLUSTRATIVE EXAMPLES

3.1 Example 1

Consider a software system having three modules with various versions of multiple alternatives for each module. The cost reliability data set is given in Table 1.

Table 1: Cost and Reliability Data for Numerical Analysis

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
3	1	0	0.001	4.5	.85	5.3	.90
	2	0	0.001	6.2	.89	4.7	.87

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. Assuming the lower bound on reliability to be Ro=0.84 and budget C = 17 units. Let the software perform three functions i.e. $L = 3,$ and the modules required to perform these function are $s_1 = \{1,2,3\}$, $s_2 = \{1,3\}$, $s_3 = \{2\}$, weights assigned to each function are $f_1 = 0.4,\ f_2 = 0.3$ and $f = 0.3$. It is also assumed that $t_1 = .01, t_2 = .05$ and $t_3 = .01$ Software package LINGO [8] is used to solve above optimization problem. The results are described as below.

The reliability and cost objectives are given weights 0.6 and 0.4 respectively. The optimal solution so obtained is

$$x_{111} = x_{122} = x_{131} = 1$$

$$x_{211} = x_{222} = x_{231} = x_{241} = 1$$

$$x_{311} = x_{323} = 1$$

It is observed that two or more alternatives are chosen for each module. Redundancy is not allowed for any module. The system reliability for the above solution is 0.8458892 and cost is 15.60007units.

3.2 Example 2

$$x_{111} = x_{123} = x_{131} = 1$$

$$x_{211} = x_{222} = x_{231} = x_{241} = 1$$

$$x_{311} = x_{323} = 1$$

It is observed that two or more alternatives are chosen for each module. Redundancy is not allowed for any module. The system reliability for the above solution is 0.877243 and cost is 17.10005 units.

Suppose management is not satisfied with reliability of system obtained in the above example. They desire to achieve more system reliability (say Ro=0.90) with the same specified budget Co = 17 units or even can compromise on the budget. It can be noted here that not all the budget is exhausted in this solution. Hence for some other set of weights which assigns even more weight to the reliability objective, a different solution can be obtained with different selection of components. Let us assign weights 0.8 and 0.2 respectively to the reliability and cost objectives, other data remaining the same. The problem now becomes infeasible; hence we solve the problem following the goal programming problem (P5). The optimal solution so obtained is

4. CONCLUSION

In this paper, Bi-Criteria optimization problem related to Component Selection for COTS based Software System under Consensus Recovery Block Scheme. Simultaneous maximization of reliability and minimization of cost are considered as the two objectives. Modular software performing a specified set of functions is considered. The problem is solved with illustrative examples. Criterion vector approach is used to solve the problem if a feasible solution for the problem exists with in specified bounds otherwise a goal programming approach can be used to solve the problem to obtain a compromised solution. The optimal solution provides the information about the components to be chosen for each module and level of redundancy that can be implemented at each level.

5. REFERENCES

[1] Belli F and Jadrzejowicz P. "An approach to reliability optimization software with redundancy". IEEE Trans. Soft Engg., 17/3(1991) 310-312.

[2] Berman O and Dinesh Kumar U. "Optimization models for recovery block scheme". Eur.J.Opl.Res. 115/2(1999) 368-379.

[3] Dinesh Kumar U. "Reliability analysis of fault tolerant recovery block". OPSEARCH, 35/2 (1998) 281-294.

[4] Geoffrion, AM "Proper efficiency and theory of vector maximization", Journal of .Mathematical Analysis and Application, 22(1968) 613-630.

[5] Kapur PK, Bardhan AK and Jha PC. "Optimal reliability allocation problem for a modular software system " OPSEARCH, Vol. 40, No.2, 2003

[6] Saaty TL. "How to make decision: The analytic hierarchy process". Eur.J.Opl Res., 48 (1990) 9-26.

[7] Steuer, RE. "Multiple Criteria optimization: theory, computation and application", Wiley, New York, 1986.

[8] Thiriez H. "OR software LINGO". Eur. J. Opl. Res., 124 (2000) 655-656.

[9] Vigder M. architecture for COTS based software systems". NRC report 41603, National Research Council, Canada (1998).

6. ABOUT AUTHOR'S

Deepak Kumar is Assistant Professor in Amity Institute of Information Technology, Amity University. He is Ph.D., M.S., B.S. from Delhi University. He has delivered Invited Talk in Shahid Chamran University, Iran, and published/presented more than 20 research papers in various International & National journal/Conferences. He was Joint Secretary of Organizing Committee of 4th International Conference in Quality, Reliability and Infocomm Technology (ICQRIT'2009 He has filled patent in Software Engineering. He has authored a book "Software Reliability Engineering – A Brief Description" (ISBN No. 978-3-8454-0939-9).). He is life time member of IACSIT (International Association of Computer Science and Information Technology) and SREQOM.

P. C. Jha is Associate Professor of Department of Operational Research, University of Delhi. He is Ph.D. from Delhi University. He has author and co-author of over 50 research papers published in National and International journals including conference proceedings. He is secretary of SREQOM. He is life time member of ORSI, SREQOM.

P. K. Kapur is Professor of Department of Operational Research, University of Delhi. He is Ph.D., M.S., B.S. from Delhi University. He has visited abroad and India extensively. He has author and co-author of over 250 research papers published in National and International journals including conference proceedings. He war former President of Operational Research Society of India. He has authored several books. He is life time member of ORSI, SREQOM.

U. Dinesh Kumar is Professor of Indian Institute of Management, Bangalore. He is Ph.d. from IIT, Bombay. He has He has author and co-Author of over 75 research papers published in National and International journals including conference proceedings. He has authored several books. He has given best Young Teacher by Association of Indian Management Schools. He is received visiting Fellowship from Queensland University of Technology. He is secretary of SOLE - The International Society of Logistics, Southwest Chapter, England. . He is life time member of ORSI, SREQOM.