韓國 OA學會 論文誌
第3卷 第2號, 1998. 6.
98-3-2-1-2

소프트웨어 제품 품질을 위한 단일 자원 할당 모델

윤 민 석* 김 신 중**

A Unitary Resource Allocation Model for Software Product Quality Project

Min-Suk Yoon* Shin-Joong Kim**

Abstract

In this paper, we develop three context-dependent resource allocation models to maximize user satisfaction in terms of software quality. Those models are formulated on the grounds that human resources are dominant in a software development project, while taking into account managerial quality constraints of the system. The satisfaction function on exerted resources plays a key linking pin between the two sides, and its functioning forms bring about different solution methods. In addition to a basic linear model, an extension model is formulated so that it may be applicable to the situation of multiple-goal settings. Finally, non-linear model is given the solving optimization algorithm developed and proved in this paper.

요 약

본 연구는 소프트웨어의 사용자 품질수준 극대화를 위한 세 가지 자원할당 모델을 제시한다. 제시되는 모델들은 소프트웨어 개발 특성상 가장 중요시되는 인적자원과 각 품질요소의 관리적 제약수준을 고려하고 있다. 각 모델의 최적 자원할당 해법은 할당된 자원량과 제품품질 수준의 관계를 규정하는 효용함수의 형태에 따라 다르다. 본 연구는 효용함수의 형태를 선형과 비선형 모델로 구분하며 또한 선형 모델은 기본모델과 확장모델로 구별하여 제시한다. 각 모델에 따른 최적 자원해법을 제시하고 비선형 모델의 경우 최적해임을 수리적으로 증명한다.

* 고려대 경영학과 강사
** 대진대 경영학과 전임강사

논문접수 : 98.4.10 심사완료 : 98. 6.15

1. Introduction

The recent studies on software quality have focused on the evaluation and improvement of software quality from the view of product or process, whereas one of the managerial concerns in a real situation is allocating scarce resources among many activities on the development life cycle of a software project. Accordingly, software resource allocation model is required as a linking pin between product quality and development activities.

Quality is a driver for user satisfaction, that is, producing a quality product is an important part of ensuring customer satisfaction [24]. Fully satisfied customers lead to a stronger competitive position resulting in higher market share and profit [17]. Thereby, mapping customer needs to product design is becoming the central issue in product development [9]. The dominant view of the mapping is on the belief that good quality of development processes assures good quality of products, which is no exception in software quality [13].

Software development generally progresses through analysis, design, coding, testing and maintenance [22] and each phase is comprised of specific activities [20]. Resource allocation is viewed in leveraging product design to enhance customer satisfaction [17].

In brief review of recent studies on resource allocation in information systems fields, Zahedi and Ashrfi [26] proposes a model to determine how reliable software modules and programs must be to maximize the user's utility, while taking into account the financial and technical constraints of the system. Cheng et. al. [2] present the optimal allocation and backup of computing resources in a multinational firm with transforming a game-theoretic model to linear programming problem. However, those studies are insufficient to reflect the high point quality view of users, and inflexible to apply to different utility requirements. Jung and Yoon [16] proposes a conceptual model to maximize user satisfaction level of a software which is also solved by linear programming. We intend to develop managerial context models based on two dominant utility form, linear and logarithmic, so that their application may be dynamic and flexible to the software developers in real situations.

The rest of this paper is organized as follows; Section II identifies the framework of software resource allocation to enhance end-user satisfaction in terms of quality and general aspects of multi-criteria resource allocation.

The following two sections proposes mathematical statements of software resource allocation and solution methods according to linear utility function and logarithmic utility function respectively. Each of the two sections also illustrates example(s) of the model application. The final remarks closes this paper with future research directions.

2. Software Resource Allocation

Software quality is the inherently multi-dimensional concept that has been discussed through the hierarchical quality model from the review of previous research [21]. A software quality model is the set of characteristics and the relations between them which provide the basis for specifying quality requirements and evaluating quality [15]. On the other hand, the optimal allocation of human resources is a dominant concern to software project managers, which is characterized unitary resources in software development. The most important contributor to a successful software project is a human factor [3] and the cultivation of motivated, highly skilled software people has been discussed since 1960s [22]. The Software Engineering Institute has developed a people management capability maturity model to enhance software development capability [4]. Accordingly, melding software quality and resources deployment drives the managers to select the best policy of resource allotment to each activity.

Developing quality product with given amount of human resources arrives at multi-criteria resource allocation (MCRA) problems. The useful decision making techniques for solving MCRA problems converts itself into an equivalent single objective maximization-type problem subject to given constraints. Ramanathan and Ganesh [23] identified two approaches to this problem. One approach is to use the priorities of confliction criteria as coefficients in a single objective into maximization-type linear programming (LP) problem. The other approach is based on the benefit-cost ratios of each activity. Goal programming (GP) is also used for resource allocation in a multiple objective environment [18].

The objective function in this study may well be multi-attribute utility (MAU) form that are designed to obtain the utility of quality characteristics as criteria. The utility is the function the amount of resources to be allocated to activities for each quality characteristic. In addition to unitary human resource constraint, our model consider managerial upper and lower bounds to be more realistic. Thus, the allocation model has the following basic structure:

objective : $$\max \ U = \sum_{j=1}^{n} w_j u(x_j) \qquad (1)$$

subject to : $$\left.\begin{aligned} & u(x_j) \geq LB_j \quad \text{for } \forall_j \\ & u(x_j) \leq UB_j \quad \text{for } \forall_j \\ & \sum_{j=1}^{n} x_j \leq B_0 \\ & x_j \geq 0 \qquad \text{for } \forall_j \end{aligned}\right\} \qquad (2)$$

where, $u(x_j)$: some algebraic satisfaction function of x_j

x_j : resources to be allocated for the jth characteristic

w_j : the weight of the jth characteristic

n : the number of characteristics to be considered

UB_j : managerial upper bound of the jth characteristic

LB_j : managerial lower bound of the jth characteristic

B_0 : total amount of resources.

The first constraint in (2) is the managerial constraint. UB_j is the highest possible level for satisfaction of characteristic j. Regardless of how much effort is put into software development process, there is an upper limit as to the quality that can be attained [1]. LB_j is the level below which the satisfaction of the jth characteristic must not fall.

For example, software reliability requirements can be expressed in a number of ways, of which the simplest, perhaps, is to impose a minimum acceptable level of reliability [26]. These two levels are the managerial limits that the project manager should determine at the planning or design stage of the software development. For the second constraint in (2), the given B_0 might be expressed in total amount of man-hours or like. The characteristics to be considered is different from the software product and evaluation purposes [14], which determines the number n.

The solution of the basic structure depends on the type of $u(x_j)$. Simple case utility function is linear form and the alternative is logarithmic form [12]. Both the alternative forms of MAU models and the alternative methods for obtaining model parameters have been studied quite extensively by researchers in psychology, economics, and decision theory. Fishburn [6, 7] published two well-known review articles on the mathematical aspects of MAU models. This study addresses unique solution method according to three different contexts, linear form of $u(x_i)$ and its extension with multiple goals, and logarithmic form of $u(x_i)$ in the following two sections.

3. Linear Function Model

3.1. Linear Function with Fixed Cost

Linear $u(x_i)$ represents user satisfaction level for each quality characteristic increase by the linear coefficient (a_i) as the amount of x_j is increasingly invested. In general, fixed costs (F_i) are all or nothing depending on allocated resources, which indicates that 0-1 integer programming may be appropriate as follows:

$$u(x_i) = \begin{cases} a_i\,(x_i - F_i) & \text{if } x_i > F_i \\ 0 & \textit{otherwise} \end{cases}.$$

However, if all LB_js are greater than zero and resources are enough to satisfy thes $u(x_i) = a_i\,(x_i - F_i)$e condition, then $x_j \geq F_j$ and . 0-1 integer variable is not necessarily involved. A mathematical statement of (1) and (2) can be transformed as follows:

objective :

$$\begin{aligned} \max\ U &= \sum_{j=1}^{n} w_j a_j(x_j - F_j) = \sum_{j=1}^{n} \theta_j(x_j - F_j) \\ &= \sum_{j=1}^{n} \theta_j x_j - \sum_{j=1}^{n} \theta_j F_j \end{aligned} \tag{3}$$

subject to :

$$\left.\begin{aligned} a_j(x_j - F_j) &\geq LB_j \quad \text{for } \forall_j \\ a_j(x_j - F_j) &\leq UB_j \quad \text{for } \forall_j \\ \sum_{j=1}^{n} x_j &\leq B_0 \\ x_j &\geq 0 \qquad \text{for } \forall_j \end{aligned}\right| \tag{4}$$

where, θ_j is adjusted weight of w_j by a_j.

Since the objective function and all the constraints are linear form, and there are no integer variables, this problem can be obviously solved by LP. However, if the objective function of (3) is looked into carefully, it can be separated into two terms in its last equation. Since θ_j and lz are given, the second term is a constant. Therefore we would be interested in only the first term, and an optimal solution could be obtained by the following procedural steps.

step 1 : Increase x_j until $a_j\,(x_j - F_j) = LB_j$ for $\forall_j$.

step 2 : Increase x_j until $a_j\,(x_j - F_j) = UB_j$ by descending order of θ_j.

step 3 : Go to step 2 until $\sum_{j=1}^{n} x_j = B_0$.

3.2. Extended Model with Multiple Goals

For the linear model of (3) and (4) (hereafter referred to as basic linear model), if the given resources cannot reach the amount to satisfy all the minimum requirements(LB_js), the basic linear model would be terminated with an infeasible solution. In order to solve this problem, some action should be exerted like setting priorities to constraints; for example, (1st) not exceeding given budget constraint, (2nd) meeting the lower bound in order, and (3rd) maximizing product quality level. If there exist some priorities of constraints, then the constraints become goals [11].

Turning to another situation, human satisfaction factors are classified into two categories, motivator and hygiene factor [10]. Similar concepts are shown in the consumer satisfaction as satisfier and dissatisfier [5]. Applying those concepts to software quality, for instance, though reliability is a significant factor, once an acceptable level of reliability is achieved, other factors dominate [17]. Thereby, software quality characteristics may be categorized into two classes of characteristics as follows.

- Satisfier : the characteristics that gives the condition such as $u(x_j) \geq LB_j > 0$, i.e., it has minimum requirement level and it should increase if possible.
- Dissatisfier : the characteristics that gives the condition such as $u(x_j) = T_j > 0$, i.e., it has target level (T_j) but not necessarily beyond the level.

Since this kind of problem has priorities among goals including lower bound constraints, it is so called a problem with muti-goals. Sometimes, preemption level would be involved. We introduce goal programming (GP) to solve the problem. The formulation is given by:

$$\text{objective : } \min Z = \sum_{k=1} \sum_{i=1}^{n_k} p_k(v_{ik}^- d_{ik}^- + v_{ik}^+ d_{ik}^+) \quad (5)$$

subject to :

$$\left.\begin{array}{ll} \sum_{j=1}^{n} \theta_j(x_j - F_j) - d_1^+ + d_1^- = 100 & \\ \sum_{j=1}^{n} x_j - d_2^+ + d_2^- = B_0 & \\ a_j\,(x_j - F_j) - d_j^+ + d_j^- = T_j & \text{for } j \in D \\ a_j\,(x_j - F_j) - d_j^+ + d_j^- = LB_j & \text{for } j \in D^c \\ x_j,\ d_i^+,\ d_i^- \geq 0 & \text{for } \forall\ i \text{ and } j \end{array}\right\} \quad (6)$$

where, D/Dc : the set of Dissatisfiers/Satisfiers,

n(D)+n(Dc) $= n$

p_k : the kth preemptive level

n_k : the number of goals assigned with a same p_k, $\sum_k n_k = n+2$

d_{ik}^+ / d_{ik}^- : positive/negative component of deviation from ith goal of a p_k

v_{ik}^+ / v_{ik}^- : weight or penalty for d_{ik}^+ / d_{ik}^-

The first equation of constraints (6) is from the objective function in the basic linear model, and its right hand side represents maximum quality value, 100%. The second equation of (6) is corresponding to the third of constraints (4), and the third and the forth of (6) are to the first and the second of (4) respectively.

3.3. Example 1 : Comparison of two linear models

For the illustrative example, we adopt the case of a software package from Yoon's study [25] where five characteristics were selected and their weights were obtained (slightly adjusted in this paper) by judgments of end-users. The rest of necessary information is assumptively given as shown in Table 1. Let Functionality, Usability and Portability be satisfiers, and Reliability and Efficiency be dissatisfiers. The preemptive goal levels are given as 1) not to violate the resource constraint, 2) to satisfy the minimum requirements and target levels, and 3) to maximize the total amount of user satisfaction level. All the necessary upper bound limits are simply assumed 100%. We have compared this extended linear model with the basic linear model in two contexts of resources.

case 1) B_0=200 : Both methods yield the same optimal solution as follows:

$(x_1, x_2, x_3, x_4, x_5)$ = (60, 40, 20, 40, 40), objective function value Z = 87.

case 2) B_0=180 : The basic linear method is terminated with an infeasible solution because all the minimum requirements can not be satisfied with the given budget. But the extended method of

Table 1. Information about Example 1

Set	Characteristic (x_j)	w_j	a_j	F_j	θ_j	T_j/ LB_j
set D^c	Functionality (x_1)	0.30	2	10	0.60	90
	Usability (x_2)	0.20	3	10	0.60	80
	Portability (x_3)	0.10	4	5	0.40	60
set D	Reliability (x_4)	0.25	3	10	0.75	90
	Efficiency (x_5)	0.15	2	5	0.30	70

GP find a solution as follows:

$(x_1, x_2, x_3, x_4, x_5)$ = (43.3, 36.7, 20, 40, 40),

objective function value Z = 75.

As shown in the example, the method of GP is more flexible than the method of LP in the aspect that setting priorities can be changed.

4. Non-Linear Function Model

4.1. Formulation and Otptimization Algorithm

In this section, the user satisfaction is described as marginally decreasing by resources as concave utility function. The function is assumed logarithm form, i.e., $u(x_j) = a_j \ln x_j$, of Huber [12]. The mathematical formulation of the resource allocation to maximize software quality can be written as follows:

$$\text{objective : } \max\ U = \sum_{j=1}^{n} w_j a_j \ln x_j = \sum_{j=1}^{n} \theta_j \ln x_j \quad (7)$$

$$\text{subject to : } \left.\begin{array}{ll} a_j \ln x_j \geq LB_j & \text{for } \forall_j \\ a_j \ln x_j \leq UB_j & \text{for } \forall_j \\ \sum_{j=1}^{n} x_j \leq B_0 & \\ x_j \geq 0 & \text{for } \forall_j \end{array}\right| \quad (8)$$

where, θ_j is adjusted weight of w_j by a_j.

Prior to describing the optimization rule, three values are denoted in the rule as

$\underline{x_j}$: the value of x_j that satisfies $a_j \ln \underline{x_j} = l_j$,

$\overline{x_j}$: the value of x_j that satisfies $a_j \ln \overline{x_j} = u_j$,

x_j^* : the current optimal value of x_j in the process of optimization.

step 1 : Construct a set of all n variables (called set S_0) and increase x_j until $\underline{x_j}$, and $x_j^* = \underline{x_j}$, for $\forall_j$

step 2 : B = $B_0 - \sum_{all\ j} x_j^*$. If B=0 or $S_0 = \{\varnothing\}$, then stop,

step 3 : Construct a set of immediately considering variables (called set S) whose element(s) is/are such j as $\max_j \{ \frac{\theta_j}{x_j^*}$ for j$\in S_0$, where $\frac{\theta_j}{x_j^*}$ is the gradients of $u(x_j)$ (= $\frac{\partial U}{\partial x_j}$) at $x_j = x_j^*$}.

step 4 : Determine bound value of gradient (b^*) by

$$b^* = \max\{ \frac{\theta_j}{x_j^*} \text{ for } j \in S_0 \text{ and } j \notin S \}.$$

step 5 : For k$\in S$, $y_k = \frac{\theta_k}{\sum_{k \in S} \theta_k} B$ and

$$x_k^* = \min\{ x_k^* + y_k, \frac{\theta_k}{b^*}, \overline{x_k} \}.$$

If $x_k^* = \overline{x_k}$, then delete variable k from the set S_0. Go to step 2.

This rule guarantees optimal solution. The optimal solution under the given (7) and (8) is mandatory according to the Slater's constraint qualification because the objective function is concave over the convex set of constraints (Proofs are trivial and well described in [19], p. 116). The concavity of the objective function leads to the following proposition 1.

Proposition 1 :

Without upper and lower bounds at initial state, all $\partial U/\partial x_j$ at $x=x_j^*$ are equally constant, and $\sum_{all\ j} x_j^* = B_0$.

Intuitively it is natural that resource constraint be strictly binding because of all increasing functions of resources. The equal gradients implies the optimal solution of x_j (x_j^*) is commensurate with relative effect of θ_j (in fact, $\theta_j/\sum_{all\ j}\theta_j$). The optimization rule with upper and lower bounds is developed from the idea to preserve equal partial gradients of all utility functions as possible.

Proposition 2 :

From the point where $\theta_k/x_k^* = c$ for k$\in S$ and k$\geq$2, resulted from step 3, assigning additional resources by $y_k=\theta_k \cdot B/\sum_{k\in S}\theta_k$ at step 5, preserves equal partial gradients, $\theta_k/(x_k^*+y_k)$ for k$\in S$.

If any $\theta_k/(x_k^*+y_k) < b^*$ for k$\in S$, same inequalities are applied to all k$\in S$. This case makes increment binding to θ_k/b^*, which includes, in the next iteration, the variable that gives b^*. The variable that $x_k^*+y_k \geq \overline{x_k}$ is no more necessarily considered because the gradients of the other variable never exceed $\theta_k/\overline{x_k}$ with incremental resources.

Proposition 3 :

If the variable that gives b^* in immediately before iteration is included in S at step 3, the improved solution at step 5 is optimal providing any value of x_k for k$\in S$ is not binding to θ_k/b^*or $\overline{x_k}$.

By iterative induction of proposition 3, we can get to the optimal point exhausting given resources (B_0).

4.2. Example 2

We illustrate a problem similar to that of example 1, given B_0=200. Minor differences are coefficient of utility function (a_j) and upper bound (u_j). There are also 5 characteristics considered for a software package and the necessary information is given in Table 2. Consecutively, Table 3 shows the processes and result of each iteration of proposed optimization algorithm. Solution that meet lower and upper bounds are as follows;

$(\underline{x_1}, \underline{x_2}, \underline{x_3}, \underline{x_4}, \underline{x_5})$=(90.0, 14.4, 4.5, 20.1, 33.1)

$(\overline{x_1}, \overline{x_2}, \overline{x_3}, \overline{x_4}, \overline{x_5})$=(148.4, 28.0, 12.2, 28.0, 148.4)

5. Conclusion with Final Remarks

In this paper, we argue for making the resource allocation a linking pin between the software development project and the high point quality, i.e., user satisfaction in terms of quality. The most important resources related to software development has been known as human resources which was unidimensional input element in this paper. There are constraints in maximizing software quality: managerial considerations of upper and lower bounds in each quality characteristic, and the finite amount of

Table 2. Result of Example 2

Iteration	B	set S_0	set S	$(x_1^*,\ x_2^*,\ x_3^*,\ x_4^*,\ x_5^*)$	Z
1	37.9	{1, 2, 3, 4, 5}	{3}	(90.0, 14.4, 9.6, 20.1, 33.1)	85.0
2	36.8	{1, 2, 3, 4, 5}	{2, 3}	(90.0, 16.1, 10.7, 20.1, 33.1)	86.2
3	30.0	{1, 2, 3, 4, 5}	{2, 3, 4}	(90.0, 28.0, 12.2, 28.0, 33.1)	92.5
4	8.7	{1, 5}	{5}	(90.0, 28.0, 12.2, 28.0, 41.8)	93.2
5	0.0	Stop			

Table 3. Information about Example 2

Characteristic (x_j)	w_j	a_j	θ_j	LB_j	UB_j
Functionality (x_1)	0.30	20	6.0	90	100
Usability (x_2)	0.20	30	6.0	80	100
Portability (x_3)	0.10	40	4.0	60	100
Reliability (x_4)	0.25	30	7.5	90	100
Efficiency (x_5)	0.15	20	3.0	70	100

resources. Maximizing the user satisfaction by the optimal resource allocation depends on the forms of utility (representing user satisfaction) function. This paper concentrated on three different contexts originating from two different utility function forms, linear and non-linear. The two solution method with the first two context under linear form resulted in

LP and GP respectively. The solution method of non-linear function for the remaining context was developed and its optimality proved in this paper. We showed an example of how the model could be applied and briefly discussed its implications.

The expanding market and the intensifying competition of software industry is drawing attention to the development project of high quality software. However, the area of resource allocation is neglected. Our model is an attempt to remind ones this issue and to open a new thinking in the amalgamation of two different view points from users and developers. This paper leaves the empirical proof of non-linear utility function in this area to the future.

References

[1] Brown, D.B., S. Maghsoodloo, and W.H. Deason, "A Cost Model for Determining the Optimal Number of Software Test Cases", IEEE Transactions on Software Engineering, Vol. 15, No. 2, pp. 218-225, 1989.

[2] Cheng, H.K., M. Freimer and U. Sumita, "Optimal Allocation and Backup of Computer Resources under Asymmetric Information and Incentive Incompatibility," European Journal of Operational Research, Vol 91, No. 2, 1996, pp. 411-426.

[3] Curtis, B., "A Field Study of the Software Design Process for Large Systems," IEEE Transactions on Software Engineering, Vol. 31, No. 11, 1988, pp. 1268-1287.

[4] Curtis, B., People Management Capability

Maturity Model, Software Engineering Institute, Pittsburgh, PA, 1994.

[5] Donnelly, Jr., J. H., "Customer cannot be satisfied until after they are not dissatisfied", Bank Marketing, January, 1991.

[6] Fishburn, P, "Methods of Estimation Additive Utilities", Management Science, Vol. 13, No., 7, 1967.

[7] Fishburn, P, "Utility Theory", Management Science, Vol. 14, No., 5, 1968.

[8] Garvin, D.A., Managing Quality: The Strategic and Competitive Edge, The Free Press, New York, 1988.

[9] Hauser, J.R. and D.P. Clausing, "The House of Quality," Harvard Business Review, Vol. 66, No. 3, 1991, pp. 63-73.

[10] Herzberg, F, B. Mausner and B. Synderman, "The Motivation to Work," John Wiley and Sons, New York, 1959.

[11] Hiller, F.S. and G.J. Liebernam, "Introduction to Operations Research", 5th. edition, McGrawHill, New York, 1990.

[12] Huber, G., "Multi-Attribute Utility Models: A Review of Field and Field-Like Stduies", Management Science, Vol. 20, No. 10, pp. 1393-1402, 1974.

[13] Humphrey, W.S., Managing the Software Process, Addison-Wesley, 1989.

[14] ISO/IEC (draft) 14598-1, Information Technology -Software Product Evaluation- Part 1 : General Overview, ISO, May, 1996.

[15] ISO/IEC 9126 Information Technology-Software Product Evaluation- Quality Characteristics and Guidelines for Their Use, ISO, 1991.

[16] Jung H.W. and M.S. Yoon, "A Software Product Quality Evaluation and Resource Allocation Model", 5th. European Conference on Software Quality, Conference Proceedings, 1996, pp. 286-294.

[17] Kekre, S., M.S. Krishnan and K. Srinivasan, "Drivers of Customer Satisfaction for Software Products: Implications for Design and Service Support", Management Science, Vol. 41, No. 9, 1995, pp. 1456-1470.

[18] Khorramshahgol, R. and A.A. Okoruwa, "A Goal Programming Approach to Investment Decisions - A Case-Study of Fund Allocation among Different Shopping Malls," European Journal of Operational Research, Vol 73, No. 1, 1994, pp. 17-22.

[19] Luenberger D.G., Introduction to Linear and Nonlinear Programming, Addison Wesley, 1973

[20] Mosley, D. J., The Handbook of MIS Application Software Testing, Yourdon Press, New Jersey, 1993.

[21] Park, H. I., Stduy on the /W Quality Evaluation Model Using the AHP-Developing a New Judgment-Aggre
gation Method, Thesis for the Degree of Doctor, Graduate School of Korea University (Korean), 1997.

[22] Pressman, R.S., Software Engineering; A Practitioner's Approach, McGraw-Hill, Singapore, 1997.

[23] Ramanathan, R. and L.S. Ganesh, "Using AHP for Resource-Allocation Problems," European Journal of Operational Research, Vol 80, No. 2, 1995, pp. 410-417.

[24] Sanders, J. and E. Curran, Software Qaulity, Addison-Wesley, New York, 1994.

[25] Yoon, M. S., S/W Quality Evaluation Model Using the AHP - Developing a New Judgment-Aggregation Method, Thesis for the Degree of Doctor, Graduate School of Korea University, 1997.

[26] Zahedi, F. and N. Ashrafi, "Software Reliability Allocation Based on Structure, Utility, Price, and Cost", IEEE Transactions on Software Engineering, Vol. 17, No. 4, 1991, 345-356.

Appendix

Proof of Proposition 1 : Applying Lagrange multiplier λ to the third constraint of (8), the Lagrangian relaxation problem is obtained as follows:

$$L(\lambda) = \sum_{j=1}^{n} \theta_j \ln x_j + \lambda (B_0 - \sum_{j=1}^{n} x_j).$$

According to the Kuhn-Tucker optimal condition,

$$\frac{\partial L}{\partial x_j} = \frac{\theta_j}{x_j} - \lambda \le 0, \quad x_j \ge 0, \quad \text{and} \quad x_j(\frac{\theta_j}{x_j} - \lambda) = 0, \quad \text{for} \quad \forall_j,$$

$$\frac{\partial L}{\partial \lambda} = B_0 - \sum_{j=1}^{n} x_j \ge 0, \quad \lambda \ge 0, \quad \text{and} \quad \lambda (B_0 - \sum_{j=1}^{n} x_j) = 0.$$

Assuming $x_j>0$ for $\forall_j$, $\lambda = \frac{\theta_j}{x_j} > 0$, and $B_0 - \sum_{j=1}^{n} x_j = 0$ (∴ complementary slackness). (A1)

Applying $x_j = \frac{\theta_j}{\lambda}$ to the equation (A1), we obtain following solution

$\lambda = \sum_{j=1}^{n} \theta_j \Big/ B_0$, and $x_j^* = \theta_j \cdot B_0 \Big/ \sum_{j=1}^{n} \theta_j > 0$ (consistent with the assumption).

All partial gradients of $\frac{\partial U}{\partial x_i} = \frac{\theta_j}{x_j}$ at $x_j = x_j^*$ are equal to $B \Big/ \sum_{j=1}^{n} \theta_j$ for $\forall_j$.

Q.E.D.

Proof of Proposition 2 : Since all partial gradients are equal, their reciprocals are also equal. That is,

$$\theta_k / x_k^* = c \text{ for } k \in S \quad \leftrightarrow \quad x_k^* / \theta_k = 1/c \text{ for } k \in S$$

The improved solution $x_k^* + y_k = x_k^* + \frac{\theta_k}{\sum_{k \in S} \theta_k} B.$

At this point, its partial gradient is $\frac{\theta_k}{x_k^* + \frac{\theta_k}{\sum_{k \in S} \theta_k} B}$, and its reciprocal is

$\frac{x_k^*}{\theta_k} + \frac{B}{\sum_{k \in S} \theta_k}$, where the first term is equal for $k \in S$ and the second is constant.

∴ all $\frac{\theta_k}{x_k^* + y_k}$ for $k \in S$ are equally constant.

Q.E.D.

Proof of Proposition 3 : We intend directly to show the objective function value decreases with any deviation from the suggested optimal point. Simply for two arbitral variables x_p^* and x_q^*, following inequality is to be proved;

$$\theta_p \ln x_p^* + \theta_q \ln x_q^* \ge \theta_p \ln (x_p^* - \Delta x_p) + \theta_q \ln (x_q^* + \Delta x_p), \text{ where } \theta_p / x_p^* = \theta_q / x_q^* \qquad \text{(A2)}$$

To prove the above inequality (A2) is same as to prove following inequality.

$$\theta_p \ln(x_p^* - \Delta x_p) + \theta_q \ln(x_q^* + \Delta x_p) - \{ \theta_p \ln x_p^* + \theta_q \ln x_q^* \}$$

$$= \ln\left[\left(\frac{x_p^* - \Delta x_p}{x_p^*}\right)^{\theta_p} \cdot \left(\frac{x_q^* + \Delta x_p}{x_q^*}\right)^{\theta_q} \right] \le 0 \quad \equiv \left(\frac{x_p^* - \Delta x_p}{x_p^*}\right)^{\theta_p} \cdot \left(\frac{x_q^* + \Delta x_p}{x_q^*}\right)^{\theta_q} \le 1 \quad \text{(A3)}$$

If let $\Delta x_p = \varepsilon x_p$ $(0 \le \varepsilon \le 1)$, and $\theta_p / \theta_q = x_p^* / x_q^* = \psi > 0$, then (A3) is as following;

$$(1-\varepsilon)^{\theta_p} \cdot (1+\varepsilon\psi)^{\frac{1}{\psi}\theta_p} = \left\{ (1-\varepsilon) \cdot (1+\varepsilon\psi)^{\frac{1}{\psi}} \right\}^{\theta_p} \le 1$$

Now we intend just to show $F(\varepsilon, \psi) = (1-\varepsilon) \cdot (1+\varepsilon\psi)^{\frac{1}{\psi}} \le 1$

1) $\psi \ge 1$: $F(\varepsilon, \psi) \le (1-\varepsilon\psi)^{\frac{1}{\psi}} \cdot (1+\varepsilon\psi)^{\frac{1}{\psi}} = (1-(\varepsilon\psi)^2)^{\frac{1}{\psi}} \le 1$

2) $0 < \psi < 1$: $\frac{\partial F}{\partial \varepsilon} = -(1+\varepsilon\psi)^{\frac{1}{\psi}} + (1-\varepsilon)(1+\varepsilon\psi)^{\frac{1}{\psi}-1} \le 0$ (A4)

$(\because\ (1+\varepsilon\psi)^{\frac{1}{\psi}} > (1+\varepsilon\psi)^{\frac{1}{\psi}-1}$ and $(1-\varepsilon) \le 0)$.

$$\frac{\partial F}{\partial \psi} = (1-\varepsilon)(1+\varepsilon\psi)^{\frac{1}{\psi}} \cdot \left[\frac{1}{\psi} \left\{ \frac{\varepsilon}{1+\varepsilon\psi} - \frac{1}{\psi} \ln(1+\varepsilon\psi) \right\} \right] \le 0 \quad \text{(A5)}$$

$(\because\ (1-\varepsilon)(1+\varepsilon\psi)^{\frac{1}{\psi}} \ge 0$ and $[\cdot] \le 0$, under the given condition).

Since both (A4) and (A5) are decreasing function of ε and ψ respectively, $F(\varepsilon, \psi)$ is maximized when $\varepsilon = 0$ and $\psi \to 0$.

$$\lim_{\varepsilon \to 0, \psi \to 0} F(\varepsilon, \psi) = \lim_{\varepsilon \to 0} (1-\varepsilon) e^{\varepsilon} = \lim_{\psi \to 0} F(0, \psi) = 1$$

$\therefore$ $F(\varepsilon, \psi) \le 1$ and (A-4) is proved.

For considering K variables, let optimal point $\mathbb{P} = (x_1^*, x_2^*, \cdots, x_K^*)$ and deviated point $\mathbb{Q} = (x_1^* + \Delta x_1, x_2^* + \Delta x_2, \cdots, x_K^* + \Delta x_K)$ where $\sum_{k=1}^{K} \Delta x_k = 0$. Adjusting Δx_k, we can make all elements in $\mathbb{Q}$ into paired elements, $(x_p^* - \Delta x_p,\ x_q^* + \Delta x_p)$. Along each paired element, (A3) is applied additively, resulting in no increase of objective function value.

Q.E.D.

● 저자소개

윤 민 석
고려대학교 문과대학 사학과(문학사)
고려대학교 대학원 경영학과(경영학 석 • 박사)
고려대 경영학과 강사
E-mail : msyoon@kuba.korea.ac.kr

김 신 중
대진대 경영학과 전임강사
E-mail : sjkim@road.daejin.ac.kr