A METHOD FOR EVALUATION AND SELECTION OF COMPLEX HARDWARE AND SOFTWARE SYSTEMS

Jozo J. Dujmović
Department of Computer Science
San Francisco State University
1600 Holloway Ave., San Francisco, CA 94132
jozo@cs.sfsu.edu

This paper presents the Logic Scoring of Preference (LSP) - a general quantitative decision method for evaluation, comparison, and selection of complex hardware and software systems. The LSP method is a generalization and an extension of various scoring techniques. Its mathematical background is a continuous preference logic. An extensive software support for the LSP method is available. The method can be successfully used by expert teams for evaluation, comparison, selection, and optimization of general complex systems.

1 Introduction

Evaluation, comparison, and selection of modern computer and communication systems is a complex decision problem. Manufacturers of such systems are regularly represented by experienced professional salesmen. As opposed to that, buyers of these systems are frequently less experienced and less professionally prepared for the system evaluation process. The result of such an imbalance may be a high price paid for inadequate equipment. Consequently, there is a clear practical interest in system evaluation methods.

System evaluation techniques can be either qualitative or quantitative. Qualitative techniques are usually based on a list of features to be analyzed for each competitive system. The list includes technical characteristics, costs, and other components for evaluation. After a study of proposed systems the evaluator creates for each proposal a list of advantages and a list of disadvantages. The lists summarizing advantages and disadvantages are then intuitively compared and the final ranking of proposed systems is suggested. Such an approach is obviously attractive only when the decision problem is sufficiently simple. In cases with many decision criteria it is difficult to properly intuitively aggregate a number of components affecting the final decision, and it is not possible to precisely identify minor differences between similar proposals. In addition, it is extremely difficult to justify whether a given difference in total cost is commensurate to a corresponding difference in total performance. These difficulties can be reduced by introducing quantitative components in the decision process.

The aim of quantitative methods is to make the system evaluation process well structured, relatively simple, and accurate, providing global quantitative indicators which are used to find and to justify the optimum decision. Some quantitative methods are oriented toward economic effects, trying to express all decision parameters in terms of financial indicators [7]. Two main disadvantages of such an approach are that (1) for many decision variables (e.g. various important features pertinent to an operating system, compilers, DBMS, or a software monitor) it is very difficult to exactly determine what are the global financial effects they produce, and (2) the method is assumed to work within a specific economic system, and economic systems differ from country to country reducing the general applicability of the method. Another group of quantitative methods is based on various scoring techniques [2, 6, 9, 10]. These methods quantify the decision process by computing two global indicators: (1) a global preference score, and (2) a global cost indicator for each evaluated system. These indicators are then aggregated using a quantitative cost/preference analysis. Such an approach is attractive because it is equally applicable in all economic environments.

The LSP method presented below is a generalization of existing scoring methods. The method provides a means for the development of complex criterion functions using a continuous preference logic (for mathematical details see [5]). These criterion functions can be efficiently used for the evaluation, comparison, and selection of computer and communication equipment [3], and complex software systems [1, 11]. A comprehensive presentation of the LSP method can be found in [4]. In this paper we present a complete overview of the LSP method with a minimum of mathematical details. Our goal is to emphasize the differences between the LSP method and traditional scoring techniques, and to uniformly cover all phases of the evaluation process,

2 Scoring Techniques

Traditional scoring techniques are presented in detail in [9] and compared with other alternative approaches in [8]. The basic idea is very simple. For a set of evaluated systems we first identify n relevant components (performance variables) that are individually evaluated. The results of evaluation are individual normalized scores $E_1, \ldots, E_n$, where $0 \leq E_i \leq 1$ (or $0 \leq E_i \leq 100\%$). The average score is then $E = (E_1 + \ldots + E_n)/n$. If all components are not equally important then we introduce positive normalized weights $W_1, \ldots, W_n$, which reflect the relative importance of individual components. Usually, $0 < W_i < 1, \quad i = 1, 2, ..., n$, and $W_1 + \ldots + W_n = 1$. The global score is defined as a weighted arithmetic mean:

$$E = W_1 E_1 + W_2 E_2 + \ldots + W_n E_n, \quad 0 \leq E \leq 1.$$

In simple cases this approach may be adequate. However, in the case of complex hardware and/or software systems we can have more than 100 performance variables. For example, if $n = 100$, the largest weights can be just a few percent, and the smallest weights can be completely negligible. The average weight is only 1%. There are three problems emerging in this situation:

1. It is not possible to model mandatory requirements. If $E_i = 0$ this is not going to yield $E = 0$ regardless of the level of importance of the i^{th} performance variable.

2. The contribution of component E_i to the global score is limited to W_i. The only consequence of the complete absence of the i^{th} feature (expressed as $E_i = 0$) is the reduction of the global score by W_i, which is regularly not significant.

3. If some performance variables are considered significant (and this is expressed through their weights) then less significant performance variables may have contributions that are one to two orders of magnitude less than the most significant components. This fact practically limits the value of n.

The problem of mandatory requirements cannot be solved by simply using the geometric mean:

$$E = E_1^{W_1} \cdot E_2^{W_2} \cdots E_n^{W_n}.$$

Now we have the problem that if $E_i = 0$ for the most insignificant performance variable the consequence is the same as for all other performance variables: $E = 0$. Of course, this is not acceptable.

These obvious limitations and problems related to traditional scoring techniques make them inadequate for professional evaluation of complex systems. This was the motive for the development of the LSP method.

3 Basic Concepts of the LSP Method

To evaluate a system means first to establish a criterion specifying all properties the system is expected to have, and then to determine a quantitative measure of the extent to which the system satisfies the requirements defined by the criterion. Logic Scoring of Preference is a quantitative method for the realization of complex criterion functions and for their application in evaluation, optimization, comparison, and selection of general complex systems.

The phases of an evaluation process depend on the type of the evaluated system. In the case of evaluating a complex software system (e.g. a windowed environment [1], or a data management system [11]) we are primarily interested in evaluating concepts and features of the analyzed system. In such cases some parts of the evaluation process (e.g. cost analysis, contracting, installation, and acceptance tests) can be missing. In the case of evaluating computer and communication equipment we assume all phases from an initial feasibility study to the final acceptance tests. Therefore, our presentation of the LSP method will use examples from the minicomputer/mainframe evaluation and selection process, because in such cases all evaluation phases are easily visible.

In a general case of hardware acquisition the LSP method includes the following eight major steps:

1. *Feasibility study*
2. *Specification of performance variables*
3. *Definition of elementary criteria*
4. *Specification of the preference aggregation structure*
5. *Request for proposals*
6. *Preparation of proposals*
7. *System evaluation and selection using the cost-preference analysis*
8. *Contracting, equipment installation, and acceptance test*

The above list indicates that the evaluation and selection of computer systems and similar complex systems is primarily a decision problem. Consequently, the techniques for its solution must contain subjective components. This essential property of system evaluation and selection deserves special attention since sometimes subjective components of evaluation and selection methods are misinterpreted as their drawbacks. In fact, the evaluation process is always based on a set of requirements all competitive systems are expected to fulfill. These requirements are usually derived from a set of goals to be attained by the system. The set of goals can obviously be specified only by human decision makers. Accordingly, both the goals and the corresponding requirements are specified subjectively, and there is no other rational way to initiate an evaluation procedure. In a positive sense "subjective" here simply means "based on human expertise and experience". On the other hand, if the necessary expertise is missing then the goals and requirements can be ambiguous, missing, or even wrong, and the evaluation process cannot be expected to generate correct and useful results. Such a situation is by no means an inherent shortcoming of evaluation procedures – it is merely an obvious characteristic pertinent to human decision making. It is easy to note that the quantitative evaluation techniques themselves cannot prevent the intentional specification of wrong requirements in order to obtain some desired result.

Subjective components in system evaluation need not and cannot be avoided but they should be used properly. In fact, the successful system evaluation and selection methodology must provide a correct aggregation of both subjective and objective decision components. Subjective components include both a detailed specification of all relevant system requirements, and a subjective assessment of the extent to which some requirements are fulfilled (e.g., in a computer evaluation and selection process the evaluator usually directly assesses preference scores for the quality of documentation, various features of software, the quality of training and support, etc.). At the same time, the objectively measurable decision variables (e.g. memory capacities, benchmark program run times, resource utilizations, etc.) are widely used providing the objective component of the decision process.

4 An Overview of the LSP Method

Following is a short description of all major steps of the LSP method.

4.1 Feasibility Study

The LSP method is organized as an essential part of the system life cycle. We initially assume an existing operational system which is permanently evaluated by its users. The system operates unchanged as long as its global performance level is considered satisfactory. Once the global performance level is shown to be unsatisfactory, the user initiates an action aimed at changing the existing system in order to improve the global performance level. The user normally expects a number of various benefits, but also a number of related expenses. Therefore, the first step preceding all other activities must be a classic feasibility analysis showing clearly both the expected benefits and the expected costs, and justifying the whole investment [8]. Without a convincing justification of economic and other effects no further steps should be undertaken.

4.2 Specification of Performance Variables

The potential buyer should precisely specify all goals that are expected to be achieved and then systematically define a complete set of requirements for all relevant system characteristics. Each characteristic which will be individually evaluated is called the performance variable of the evaluated system. In the case of computer equipment these characteristics can be conveniently grouped in the following four sets of performance variables:

```
1. HARDWARE
   1.1 Central processor
   1.2 Main memory
   1.3 I/O processors (channels)
   1.4 Disk memory
   1.5 Tape (archive) memory
   1.6 Main I/O devices
   1.7 Data communications and networking

2. SOFTWARE
   2.1 Operating system
   2.2 Programming languages
   2.3 Utility programs
   2.4 Data management systems
   2.5 Network software
   2.6 Application software systems

3. MEASURED PERFORMANCE AND AVAILABILITY
   3.1 Measured monoprogramming performance
   3.2 Measured multiprogramming performance
   3.3 Measured interactive performance
   3.4 Measured network performance
   3.5 Reliability and availability

4. VENDOR SUPPORT
   4.1 Training
   4.2 Maintenance of hardware and software
   4.3 Consulting services
   4.4 Documentation and informing
   4.5 Auxiliary and backup systems
```

In the majority of computer evaluation studies the number of performance variables for computer system evaluation usually varies from 40 to 120 depending on the size and cost of evaluated systems. The performance variables are derived using a hierarchical decomposition of the elements in the above list. For example, the disk memory could be decomposed as follows:

```
1.4  Disk memory
     1.4.1  Total capacity of disk memory
     1.4.2  Maximum extension of disk memory
     1.4.3  Organization of disk memory
     1.4.4  Number of independent disk accesses
     1.4.5  Disk memory speed
```

Since the obtained items cannot be further decomposed they are performance variables that can be directly evaluated. The set of performance variables typically includes memory capacities, device characteristics, measured system throughputs, response times, overheads, a variety of software features, the amount of training and consulting offered, a maintenance response time, the quality of vendor's local office, etc.

4.3 Definition of Elementary Criteria

For each performance variable $X_i, i = 1, \ldots, n$, it is necessary to define an acceptable range of values and to create a preferential ordering function called the elementary criterion. Elementary criteria are used for preference scoring purposes. The elementary criterion function G_i is a mapping of the value of performance variable X_i into the corresponding value of the elementary preference E_i. Elementary preferences are normalized, and indicate the degree of satisfaction of buyer's requirements. In this presentation we assume $0 \le E_i \le 100\%$, i.e. E_i is interpreted as the percentage of requirements satisfied by the value of X_i. So, $E_i = 100\%$ denotes a complete satisfaction, $E_i = 0$ denotes a completely unsatisfactory situation, and a partial satisfaction of requirements yields $0 < E_i < 100\%$. For example, if X_i denotes the capacity of disk memory and if the range of acceptable values is $X_i^{min} < X_i \le X_i^{max}$ then the elementary criterion function $E_i = G_i(X_i)$ should generate preferences $E_i = 0$ if $X_i \le X_i^{min}$, and $E_i = 100\%$ if $X_i \ge X_i^{max}$. For $X_i^{min} < X_i < X_i^{max}$ the resulting elementary preference should satisfy $0 < E_i < 100\%$ and it may be convenient to organize $G_i(X_i)$ as a piecewise linear function. An example showing a possible elementary criterion for evaluating the capacity of a disk memory (M) can be expressed using the following preference scale:

```
        Disk memory capacity M [GB]
3                       4                 6
+---+---+---+---+---+---+---+---+---+---+
0  10  20  30  40  50  60  70  80  90  100
        Elementary preference, E [%]
```

This criterion specifies the requirement for more than 3 GB of disk memory (if $M \le 3$ then $E = 0$), and no more than 6 GB of disk memory (if $M \ge 6$ then $E = 100\%$, i.e. the additional memory above 6 GB gets no credit, but increases the total cost).

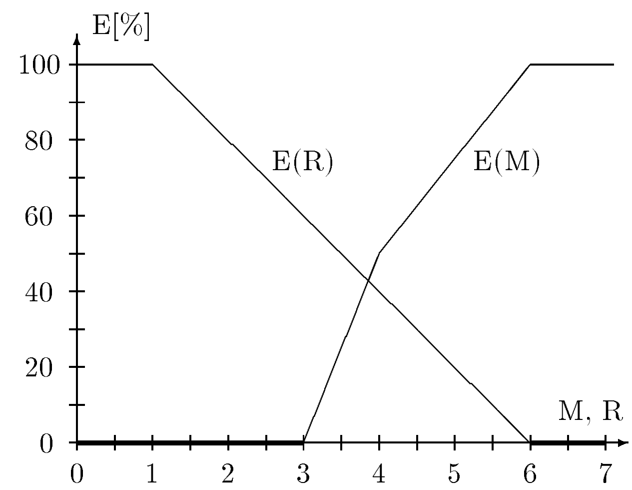

Figure 1: Sample elementary criteria

The capacity $M = 4$ GB is considered to satisfy 50% of user's requirements. Other preferences can be computed using a linear interpolation: e.g. if $M = 5$ GB then $E = 75\%$.

A similar criterion for evaluating the average response time R of a workstation can be defined using the following preference scale:

```
        Response time, R [seconds]
6                                      1
+---+---+---+---+---+---+---+---+---+---+
0  10  20  30  40  50  60  70  80  90  100
        Elementary preference, E [%]
```

Here the measured average response time $R \geq 6$ seconds is considered unsatisfactory, and the response time $R \leq 1$ second is considered excellent. Other values can be computed using interpolation: $E = 20(6 - R)$, $1 < R < 6$. Preference scales can be interpreted in a form of curves, as shown in Fig. 1.

Whenever possible the evaluator should try to define the criterion in the form of a preference scale, and in cases where that is not possible (or not convenient) the direct preference assessment can be used (e.g. during the evaluation process, after a study of all documentation for a given compiler, the evaluator may decide that the most appropriate preference score for the quality of documentation is $E = 67\%$).

4.4 Specification of the Preference Aggregation Structure

For n performance variables the elementary criteria yield n elementary preferences $E_1, \ldots, E_n$. Using a

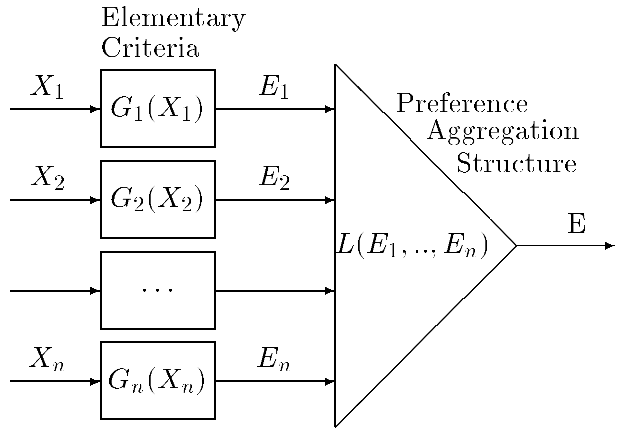

Figure 2: The LSP model of a complex criterion

stepwise aggregation technique the elementary preferences can be aggregated yielding the global preference

$$\begin{aligned} E &= L(E_1, \ldots, E_n) \\ &= L(G_1(X_1), \ldots, G_n(X_n)) \\ &= g(X_1, \ldots, X_n), \quad (0 \leq E \leq 100\%). \end{aligned}$$

The global preference E is interpreted as the global degree of satisfaction of all specified requirements. Figure 2 shows the resulting structure of the LSP criterion function.

The function L can be organized by combining weighted power means of suitably selected power: e.g. if an aggregation block has input preferences $e_1, \ldots, e_k$, and the relative significance of these inputs can be expressed using the weights $W_1, \ldots, W_k$, then the output preference e_0 may be computed as follows:

$$\begin{aligned} e_0 &= (W_1 e_1^r + \ldots + W_k e_k^r)^{1/r}, \\ & \quad W_1 + \ldots + W_k = 1, \quad W_i > 0, \quad i = 1, \ldots, k. \end{aligned}$$

The power r is a real number selected so to achieve desired logical properties of the aggregation function. The technique for selecting r is described in detail in [5] and [4]; following is a short outline of this procedure. The position of e_0 between $e_{min} = min(e_1, \ldots, e_k)$ and $e_{max} = max(e_1, \ldots, e_k)$ varies depending on the specific values of $e_1, \ldots, e_k$. Let d be an indicator of the average position of e_0 between e_{min} and e_{max} (obtained for all possible input values $0 \leq e_i \leq 1$, $i = 1, \ldots, k$). We define d so that $0 \leq d \leq 1$ and that $d = 0$ yields $e_0 = e_{min}$ and $d = 1$ yields $e_0 = e_{max}$. If $k = 2$ then the parameter r can be approximately computed from the

Table 1: Symbols and parameters of the *andor* function

Operation	Symbol	d	r2	r3	r4	r5
DISJUNCTION	D	1.0000	+infty	+infty	+infty	+infty
STRONG QD (+)	D++	0.9375	20.630	24.300	27.110	30.090
STRONG QD	D+	0.8750	9.521	11.095	12.270	13.235
STRONG QD (-)	D+-	0.8125	5.802	6.675	7.316	7.819
MEDIUM QD	DA	0.7500	3.929	4.450	4.825	5.111
WEAK QD (+)	D-+	0.6875	2.792	3.101	3.318	3.479
WEAK QD	D-	0.6250	2.018	2.187	2.302	2.384
SQUARE MEAN	SQU	0.6232	2.000			
WEAK QD (-)	D--	0.5625	1.449	1.519	1.565	1.596
ARITHMETIC MEAN	A	0.5000	1.000	1.000	1.000	1.000
WEAK QC (-)	C--	0.4375	0.619	0.573	0.546	0.526
WEAK QC	C-	0.3750	0.261	0.192	0.153	0.129
GEOMETRIC MEAN	GEO	0.3333	0.000			
WEAK QC (+)	C-+	0.3125	-0.148	-0.208	-0.235	-0.251
MEDIUM QC	CA	0.2500	-0.720	-0.732	-0.721	-0.707
HARMONIC MEAN	HAR	0.2274	-1.000			
STRONG QC (-)	C+-	0.1875	-1.655	-1.550	-1.455	-1.380
STRONG QC	C+	0.1250	-3.510	-3.114	-2.823	-2.606
STRONG QC (+)	C++	0.0625	-9.060	-7.639	-6.689	-6.013
CONJUNCTION	C	0.0000	-infty	-infty	-infty	-infty

following formula:

$$r = \rho(d) = \frac{-0.742 + 3.363d - 4.729d^2 + 3.937d^3}{d(1-d)}.$$

For $d = 0, 0.2274, 0.3333, 0.5, 0.6232, 1$ the weighted power mean reduces respectively to the pure conjunction (the minimum function, $r = -\infty$), harmonic mean ($r = -1$), geometric mean ($r = 0$), arithmetic mean ($r = 1$), square mean ($r = 2$), and the pure disjunction (the maximum function, $r = +\infty$). Thus, d is called the *disjunction degree* and the preference aggregation function

$$e_0 = (W_1 e_1^{\rho(d)} + \ldots + W_k e_k^{\rho(d)})^{1/\rho(d)}$$

can be used as a basic function in a continuous preference logic; it is called the *generalized conjunction/disjunction* and denoted *andor*.

The most frequently used values of d are $0, 1/16, \ldots, 1$. For these values it is convenient to use the special names and special symbols of the *andor* function shown in Table 1.

For $0 < d < 0.5$ *andor* is called the "quasi-conjunction" (QC) because it has properties similar to the pure conjunction. QC is used to model situations where we need a certain degree of simultaneity in satisfying the input elementary criteria and want to penalize systems that cannot *simultaneously* satisfy the input criteria. The penalizing effect can be adjusted from strong (for small values of d, from 0 to 0.25) to weak (for d close to 0.5).

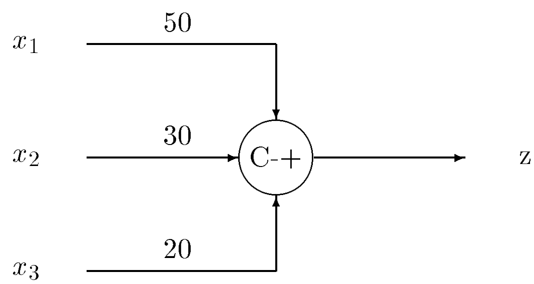

Figure 3: An example of quasi-conjunction

An example of QC is shown in Fig. 3. According to Table 1, the function C-+ in the case of three input preferences (denoted x_1, x_2, and x_3) should use the exponent $r = -0.208$. Therefore, the presented graphical notation shows the function

$$z = (0.5x_1^{-0.208} + 0.3x_2^{-0.208} + 0.2x_3^{-0.208})^{(-1/0.208)}.$$

The analysis and synthesis of such functions can be made very simple using a software tool called ANSY. For example, the sample ANSY output presented in Table 2 illustrates the properties of the analyzed function and helps the analyst to properly adjust its parameters (r, W_1, W_2, W_3).

For $0.5 < d < 1$ *andor* is called the "quasi-disjunction" (QD) because it has properties similar to the pure disjunction. QD is used to model situations where we have a certain degree of replaceability in satisfying the input elementary criteria and want to penalize only those systems that cannot satisfy *any* of the input criteria. The high level of replaceability is expressed by large values of d (from

Table 2: A sample ANSY output for the analysis of the C-+ function

```
Parameters:          r = -.208000
               W(1:3) =  .50  .30  .20
Inputs:        x(1:3) =  .70  .90  .60
Output:              y =  .730255

SENSITIVITY ANALYSIS                z(x(1),x(2),x(3))
=====================================================
x(i) =   .00 .10 .20 .30 .40 .50 .60 .70 .80 .90 1.0
-----------------------------------------------------
i = 1    .00 .25 .37 .47 .55 .61 .67 .73 .78 .83 .87
i = 2    .00 .35 .45 .52 .57 .61 .65 .68 .71 .73 .75
i = 3    .00 .47 .57 .63 .67 .70 .73 .75 .77 .79 .81
=====================================================
```

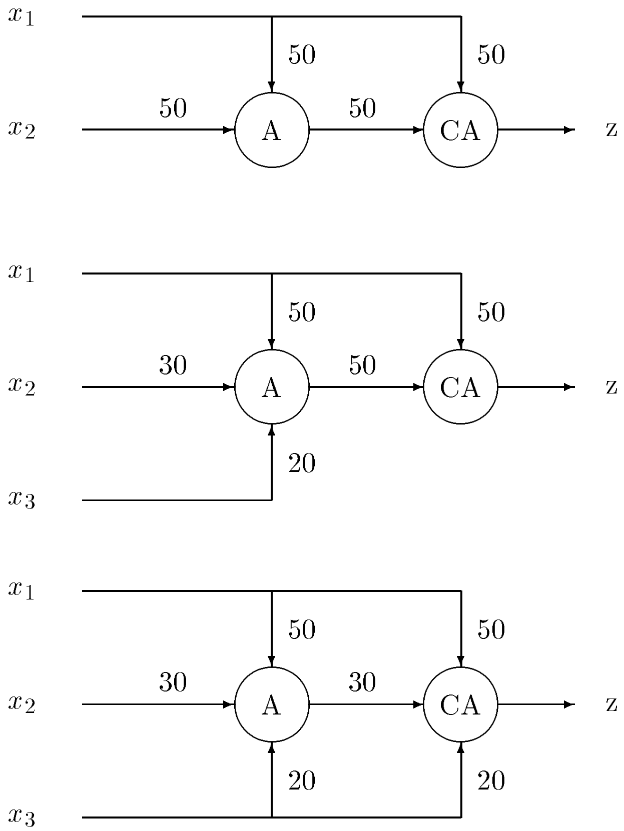

Figure 4: Conjunctive partial absorption

0.75 to 1), and a low level of replaceability corresponds to small values of d (slightly above 0.5).

For $d = 0.5$ *andor* is called the "neutrality function", because it is located right in the middle between the conjunction and the disjunction. It is used when we want to produce a criterion function having a perfectly balanced mix of conjunctive and disjunctive properties. This is the traditional (weighted) arithmetic mean.

A combination of various *andor* functions can be used to create more complex aggregation structures. The most useful such a structure is the conjunctive partial absorption exemplified in Fig. 4. The first example in Fig. 4 shows a combination of a mandatory (essential) input x_1 and a desired (or optional) input x_2. The operator CA acts as an AND gate and if $x_1 = 0$ then $z = 0$ regardless of the value of x_2. However, if $x_1 > 0$ then $x_2 = 0$ will not cause $z = 0$. It will only reduce the value of the output preference, as exemplified in Table 3.

The second example in Fig. 4 shows the case where a mandatory preference x_1 is combined with two desired/optional preferences, x_2 and x_3. Finally, the third example in Fig. 4 shows the case of two mandatory inputs (x_1 and x_3) and one desired/optional input (x_2).

Table 3: A sample ANSY analysis of a conjunctive partial absorption function

```
 P A R T I A L    A B S O R P T I O N

 x ------>-------.-------->---------.
                 |                  |
                 |W1                |W2
                 |                  |
        1-W1    .---.      1-W2    .---.
 y ----------->| q |------------->| r |------>  z
                `---'              `---'


Parameters: W1 =   .50     W2 =   .50
             q = 1.00       r = -.72
Inputs:      x =   .90      y =   .40
Result:      z =   .757614

SENSITIVITY ANALYSIS                           z(x,y)
==============================================
x =  .00 .10 .20 .30 .40 .50 .60 .70 .80 .90 1.0
z =  .00 .15 .24 .32 .40 .47 .55 .62 .69 .76 .83
----------------------------------------------
y =  .00 .10 .20 .30 .40 .50 .60 .70 .80 .90 1.0
z =  .61 .65 .69 .72 .76 .79 .82 .85 .87 .90 .92
==============================================
```

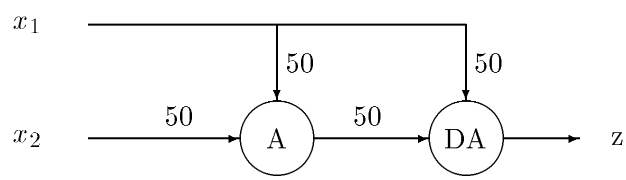

Figure 5: Disjunctive partial absorption

Another similar compound function is the disjunctive partial absorption, exemplified in Fig. 5. In this case the DA operator acts as an OR gate and enables the combination of a sufficient input x_1 and desired input x_2. Indeed, if $x_1 = 1$ then the output value z will be close to 1 regardless of the value of x_2. On the other hand, if $x_1 = 0$ then a nonzero value of x_2 can still partially compensate the lack of x_1.

The synthesis of complex criteria consists of combining elementary functions of *quasi-conjunction, neutrality, quasi-disjunction, conjunctive partial absorption* and *disjunctive partial absorption.* To illustrate the process we use examples in Figures 6 and 7. Figure 6 shows a multi-level form of a conjunctive partial absorption, where we have a fine granularity:

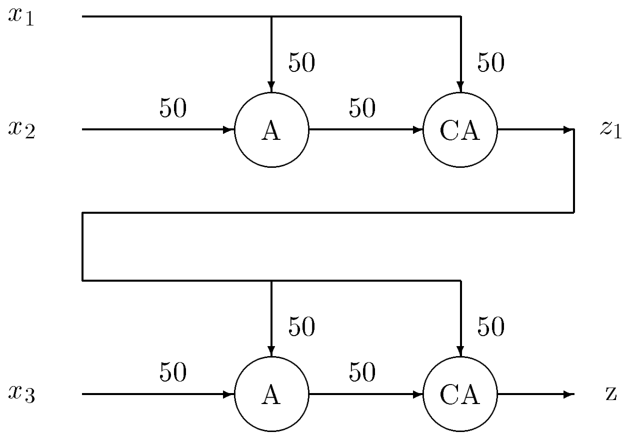

Figure 6: Multi-level conjunctive partial absorption

x_1 is mandatory, x_2 is desired, and x_3 is optional. The case in Fig. 7 combines three mandatory inputs (x_1, x_2, x_3) to generate the intermediate preference z_1. Then, z_1 is used as a mandatory input and combined with two desired/optional inputs, x_4 and x_5, to generate the output (global) preference z.

The presented examples illustrate a general method of creating complex criteria. For each group of related elementary preferences the evaluator defines an appropriate aggregation block, where the selected parameters (weights and powers) reflect user's goals and requirements. The obtained first level of aggregation blocks generates a set of aggregate preferences which can be used as inputs to the second level of aggregation blocks, forming a tree structure of aggregation blocks which eventually yields the global preference for the evaluated system as a whole. So, the function $g(X_1, \ldots, X_n)$, shown in Fig. 2, which is called the global criterion for system evaluation, must properly reflect all buyer's requirements and it should be prepared before issuing the request for proposals. Both the elementary criteria and the preference aggregation structure can be conveniently specified using appropriate LSP forms. These forms are then given to all vendors as a part of documentation which supplements the request for proposals.

4.5 Request for Proposals

The request for proposals may take any suitable legal form, and it must be accompanied by the following two additional documents: (1) System Evaluation and Proposal Preparation Guide, and, in the case of computer equipment, (2) Computer

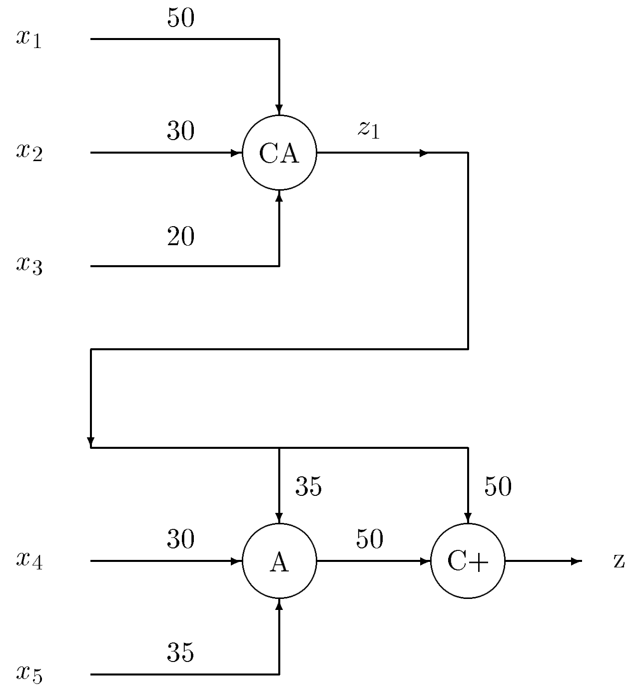

Figure 7: The case of three mandatory and two optional inputs

Performance Measurement Guide. The first document contains a complete set of buyer's requirements specified as the quantitative criterion for system evaluation in the form of LSP forms containing the set of elementary criteria and the corresponding preference aggregation structure. Computer system performance is regularly an indispensable component for evaluation and comparison of competitive systems, and accordingly the second document should specify a set of benchmark programs to be used for various performance measurements. They include monoprogrammed, multiprogrammed, interactive, and network performance measurements and their results are to be aggregated with all other system evaluation results. Therefore, using the LSP method it is possible to organize the criterion for evaluation so to get a comprehensive and complete insight into the global suitability of all evaluated systems.

4.6 Preparation of Proposals

The quantitative criterion for system evaluation precisely specifies the contents of proposals. Using the System Evaluation and Proposal Preparation Guide all vendors are given a precise and uniform guide how to prepare proposals, and they also know in advance the evaluation technique that will subsequently be applied. The vendors are asked to explicitly answer to each buyer's requirement expressed in the form of an elementary criterion. Consequently, each proposal is in fact the set of n answers to precisely specified buyer's needs and it also includes the results of performance measurements.

4.7 System Evaluation and Selection Using a Cost-Preference Analysis

Proposals prepared according to the LSP method include the values of the majority of performance variables $X_1, \ldots, X_n$ (other values are assessed by the evaluator). Using the global criterion g and an appropriate cost analysis the evaluation process eventually yields global preferences and global costs of all competitive systems. The global cost is an aggregate indicator which must reflect all financial components related to the evaluated equipment. These include the equipment purchase cost (or the total cost of equipment rental/leasing for a selected period (in some cases 4 to 7 years), all software leasing costs and all maintenance costs for the same period, software conversion costs, personnel training costs, special site preparation costs, etc. The global cost indicator C may be frequently expressed as a present value of cash flow which takes into account all possible cost components as well as the dynamics of payments [4].

If the global cost C and the global preference E of a computer system are known then a cost-preference analysis may be used to generate the resulting ranking of competitive systems. This analysis starts with the definition of limit values E^{MIN} (usually 67%) and C^{MAX} in order to reject all systems having either $E < E^{MIN}$ or $C > C^{MAX}$. The comparison of acceptable (C, E)-pairs may be based on various algorithms for computing the global quality indicator Q as a function of E and C (for details see [4]). Three sample formulas for computing Q are

$$\begin{aligned} Q &= E/C \\ Q &= pE/E_{max} + (1-p)C_{min}/C, \quad 0 < p < 1 \\ Q &= pE + (1-p)(C_{max} - C)/C_{max}. \end{aligned}$$

Here p denotes the relative significance of E, $1-p$ denotes the relative significance of C, C_{min} corresponds to the least expensive competitive system, and E_{max} corresponds to the most preferred competitive system. The acceptable systems which satisfy $E \geq E^{MIN}$ and $C \leq C^{MAX}$ can then be ranked according to decreasing values of Q or according to other similar indicators. Financial negotiations with all vendors are regularly organized in order to reduce C and to improve the value of Q. Eventually, the system with the greatest Q value should be selected as the best alternative.

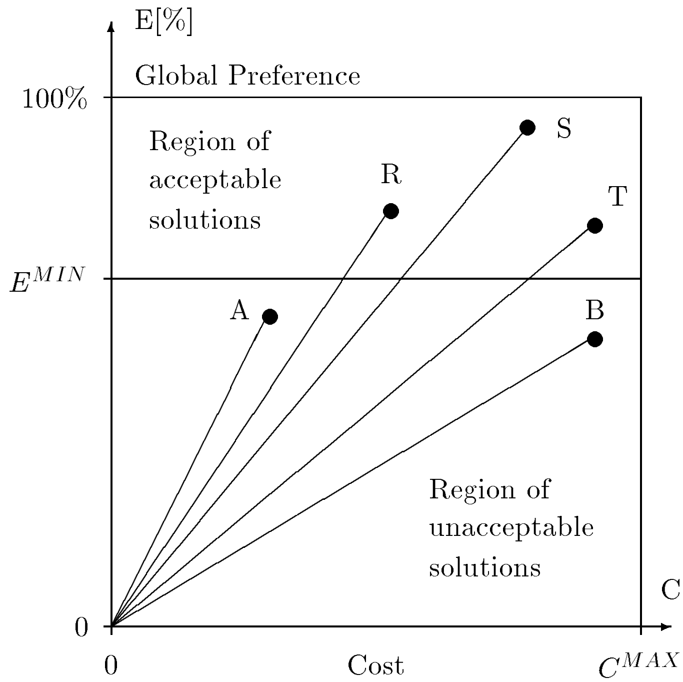

Figure 8: Cost-preference analysis

In the case of using the criterion $Q = E/C$ the cost-preference analysis is exemplified in Fig. 8. The competitive systems are A, B, R, S, and T. Using C^{MAX} and E^{MIN} we first define the regions of acceptable and unacceptable solutions. Systems A and B are rejected because they are located in the region below the E^{MIN} line. The competitive systems R, S, and T are then ranked according to the E/C ratio: the best system is R, followed by S, and T.

4.8 Contracting, Equipment Installation, and Acceptance Test

The presented LSP method substantially facilitates the preparation of contract. In fact the contract should simply include all proposed values of performance variables of the selected system, according to the list of elementary criteria. After the installation of the selected equipment it is necessary to perform the equipment acceptance test according to specifications given in the contract. Buyer's acceptance test must prove a given high level of uptime (usually 95% or more) for the complete equipment, and it must also verify all performance measurement results that the selected vendor submitted during the evaluation process. The contract must specify severe penalties for vendors using illegal techniques for improving performance measurement results and violating performance measurement conditions specified in the request for performance measurement. After the successful acceptance test the buyer may start regular operation with equipment that may be trusted to satisfy all requirements.

5 Software Support for the LSP Method

The efficiency of system evaluation process crucially depends on an appropriate software support. The standard software supporting the LSP method consists of three major systems: *ANSY*, *CDS*, and *SEL*. ANSY (ANalysis and SYnthesis of preference aggregation functions) is an interactive system which assists evaluators during the definition of the criterion function. It is used for determining optimum values of weights and powers which appear in mathematical models for system evaluation. CDS (Criterion Development System) is an interactive system which is specialized for creating and updating a data base of elementary criteria and criterion aggregation functions and for quick production of the necessary documentation supplementing the request for proposals and/or system evaluation studies. SEL (System Evaluation Language) is a specialized interpretative language used for writing programs which can evaluate, compare, and optimize the competitive systems. In addition, SEL is used for generating numerical results which are the major part of the final system evaluation report. Finally, a number of utility programs (e.g. cost analysis, cost-preference analysis, etc.) are also available and used by professional system evaluators.

6 Conclusion

The LSP method for system evaluation, comparison, and selection has the following main advantages pertinent to quantitative system evaluation methods: (1) the systematic, flexible, and complete specification of all requirements, (2) the systematic and rational evaluation process which explicitly and quantitatively shows the global level of satisfying buyer's requirements, (3) the elimination of irrational and/or illegal techniques for equipment procurement, and (4) the possibility of selecting the equipment that simultaneously optimally satisfies buyer's cost and performance criteria. In addition to the procurement of computer and communication equipment the LSP method can be used for the evaluation and comparison of a variety of other complex systems. It is particularly suitable for evaluation and validation of complex software systems, such as operating systems, graphical environments, and database systems.

References

[1] A.R. Bayucan, *Quantitative Evaluation of Windowed Environments.* M.S. Thesis. Department of Computer Science, San Francisco State University, 1996.

[2] D.R.J. White, D.L. Scott, and R.N. Schulz, *POED - A Method of Evaluating System Performance.* IEEE TEM, December 1963.

[3] J.J. Dujmović, *Computer Selection and Criteria for Computer Performance Evaluation.* International Journal of Computer and Information Sciences, Vol. 9, No. 6, pp. 435-458, 1980.

[4] J.J. Dujmović and R. Elnicki, "A DMS Cost/Benefit Decision Model: Mathematical Models for Data Management System Evaluation, Comparison, and Selection". National Bureau of Standards, Washington, D.C., No. NBS-GCR-82-374, NTIS No. PB82-170150 (155 pages), 1982.

[5] Dujmović, J.J., *Preferential Neural Networks.* Chapter 7 in "Neural Networks - Concepts, Applications, and Implementations," Vol. II. Edited by P. Antognetti and V. Milutinović. Prentice - Hall Advanced Reference Series, Prentice - Hall, pp. 155-206, 1991.

[6] T. Gilb, *Weighted Ranking by Levels.* IAG Journal, Vol. 2, No. 2, 1969, pp. 7-22.

[7] E.O. Joslin, *Computer Selection.* Augmented Edition, The Technology Press, 1977.

[8] J.P.C. Kleijnen, "Computers and Profits," Addison - Wesley, 1980.

[9] J.R. Miller III, *Professional Decision-Making*. Praeger Publishers, 1970.

[10] R.J. Quaker, *Computer Choice*. North-Holland, 1978.

[11] Su, S.Y.W., Dujmović, J.J., Batory, D.S., Navathe, S.B., and R. Elnicki, *A Cost-Benefit Decision Model: Analysis, Comparison, and Selection of Data Management Systems*. ACM Transactions on Database Systems, Vol. 12, No. 3, September 1987, pp. 472-520.