Regular Paper

Quantitative Methods for Tradeoff Analyses

Jesse Daniels,[1] Paul W. Werner,[2] and A. Terry Bahill*,[3]

[1]BAE Systems, San Diego, CA 92128

[2]High Consequence Surety Engineering, Sandia National Laboratories, Albuquerque, NM 87185-0490

[3]Raytheon Missile Systems and Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721-0020

Received May 26, 2000; accepted February 22, 2001

ABSTRACT

Practicing engineers often find the tools and techniques used for investigating alternative system designs to be cumbersome or complicated. This article will show that these systems engineering tools and techniques are in fact quite simple and can provide critical insight into how stakeholder requirements drive the engineering design process. This helps ensure that customer requirements are satisfied throughout the entire system lifecycle and aids in reducing expensive design iterations due to poorly understood or poorly documented requirements. These goals are achieved by deriving figures of merit and combining them using standard scoring functions to steer efforts towards fulfilling the customer's objectives early in the design process. Few papers in the literature capture the basic elements of tradeoff analyses in a way that entices the engineer to utilize the techniques. We have attempted to ameliorate this problem. Most of the practices presented in the literature are written from a decision analysis perspective. The success of such techniques is dependent on the expertise of the analyst in that several of the methods require considerable analyst experience for them to be employed effectively. This paper presents standardized methodologies for carrying out tradeoff analyses, which are applicable to a wide array of problems and also demonstrates that these techniques are relatively simple to use. © 2001 John Wiley & Sons, Inc. Syst Eng 4: 190–212, 2001

1. INTRODUCTION

Tradeoff studies are an important part of the systems engineering process. However, practicing engineers often find the tools and techniques for performing trade-off studies to be cumbersome or complicated. This paper will show that these systems engineering tools and techniques are in fact simple and can provide critical insight into how stakeholder requirements drive the engineering design process. This helps ensure that cus-

*Author to whom all correspondence should be addressed at: Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721-0020 (e-mail: terry@sie.arizona.edu).

Contract grant sponsor: Federal Aviation Administration; contract grant number: AAR-424

Contract grant sponsor: U.S. Department of Energy; contract grant number: DE-AC04-94AL85000

Systems Engineering, Vol. 4, No. 3, 2001

Table I. Conceptual Outline of This Paper

- Human problem solving
  - **S**tate the problem
  - **I**nvestigate alternatives
    - brain storming
    - designing alternative systems
    - modeling the systems
    - perform tradeoff analyses
      - create hierarchical structure
        - tabular example
      - assign weights of importance
      - select a method for combining data
        - linear combination
        - product combination
        - exponential combination
        - sum minus product combination
        - certainty factors
      - design figures of merit (FoMs)
        - designing FoMs
        - scoring functions
          - custom scoring functions
          - standard scoring functions
            - exponential form
            - Wymore's six
            - Wymore's first
            - The basic four
      - consider the do nothing alternative
      - conduct sensitivity analysis
  - **M**odel the system
  - **I**ntegrate
  - **L**aunch the system
  - **A**ssess performance
  - **R**e-evaluate

tomer requirements are satisfied throughout the entire system lifecycle. This paper shows techniques for doing tradeoff studies and also shows how the tradeoff process fits into the overall systems engineering process, as is shown in Table I.

2. THE SIMILAR PROCESS

Humans (individually, on teams, and in organizations) can employ simple processes to increase their probability of success. Many authors, both technical and nontechnical, have described these processes, and their descriptions are similar. Bahill and Gissing [1998] compared these processes and extracted the similarities: **S**tate the problem, **I**nvestigate alternatives, **M**odel the system, **I**ntegrate, **L**aunch the system, **A**ssess performance, and **R**eevaluate. These seven functions can be summarized with the acronym SIMILAR: **S**tate, **I**nvestigate, **M**odel, **I**ntegrate, **L**aunch, **A**ssess, and **R**eevaluate. In this section, we will briefly describe the SIMILAR process, shown in Figure 1, to give the reader an overall context for applying tradeoff studies. It is important to note that the SIMILAR process is not sequential. The functions are performed in a parallel and iterative manner.

2.1. State the Problem

The problem statement starts with a description of the top-level functions that the system must perform, or else the deficiency that must be ameliorated. The systems mandatory and preference requirements should be traceable to this problem statement [Bahill and Dean, 1999]. Acceptable systems must satisfy all the mandatory requirements. The preference requirements are traded off to find the preferred alternative. Hopefully your customer will seldom force you to tradeoff mandatory requirements. The problem statement should be in terms of *what* must be done, not *how* to do it. It might be composed in words or as a model. Inputs come from end users, operators, maintainers, suppliers, acquirers, owners, bill payers, regulatory agencies, victims, sponsors, manufacturers, and other stakeholders.

2.2. Investigate Alternatives

Alternative designs are evaluated based on performance, cost, schedule, and risk figures of merit (FoMs). No design is likely to be best on all FoMs, so multicriteria decision-aiding techniques should be used to reveal the preferred alternatives. This analysis should be redone whenever more data are available. For example, FoMs should be computed initially based on estimates by the design engineers. Then models should be constructed and evaluated. Next simulation data should be derived. Subsequently, prototypes should be measured and finally tests should be run on the real system. For

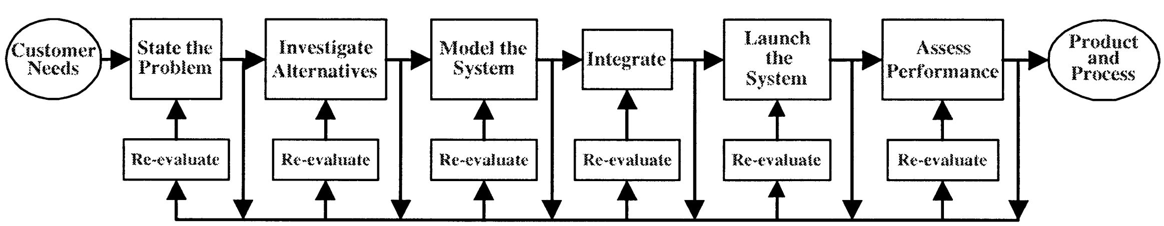


Figure 1. The SIMILAR Process from Bahill and Gissing (1998) © 1998 IEEE.

the design of complex systems, alternative designs reduce project risk. Investigating innovative alternatives helps clarify the problem statement. Most of this paper is about the Investigate Alternatives function.

2.3. Model the System

Models will be developed for most alternative designs. The model for the preferred alternative will be expanded and used to help manage the system throughout its entire life cycle. Many types of system models are used such as physical analogs, analytic equations, state machines, block diagrams, functional flow diagrams, object-oriented models, computer simulations, and mental models [Blanchard and Fabrycky, 1998]. Systems engineering is responsible for creating a product and also a process for producing it. So, models should be constructed for both the product and the process. *Process* models allow us, for example, to study scheduling changes, create dynamic PERT charts, and perform sensitivity analyses to show the effects of delaying or accelerating certain subprojects. Running the process models reveals bottlenecks and fragmented activities, reduces cost, and exposes duplication of effort. *Produc*t models help explain the system. These models are also used in tradeoff studies and risk management.

As previously stated, the systems engineering process is not sequential: It is parallel and iterative. This is another example: Models must be created before alternatives can be investigated.

2.4. Integrate

No man is an island. Systems, businesses, and people must be integrated so that they interact with one another. Integration means bringing things together so they work as a whole. Interfaces between subsystems must be designed. Subsystems should be defined along natural boundaries. Subsystems should be defined to minimize the amount of information to be exchanged between the subsystems. Well-designed subsystems send finished products to other subsystems. Feedback loops around individual subsystems are easier to manage than feedback loops around interconnected subsystems.

2.5. Launch the System

Launching the system means running the system and producing outputs. In a manufacturing environment this might mean buying commercial off-the-shelf hardware and software or it might mean actually making things, e.g., bending metal. Launching the system means allowing the system to do what it was intended to do.

2.6. Assess Performance

FoMs, technical performance measures, and metrics are all used to assess performance. FoMs are used to quantify requirements in the tradeoff studies. Technical performance measures are used to mitigate risk during design and manufacturing. Metrics are used to help manage a company's processes. Measurement is the key. If you cannot measure it, you cannot control it. If you cannot control it, you cannot improve it [Moody et al., 1997].

2.7. Re-evaluate

Re-evaluation is arguably the most important of these functions. For a century, engineers have used feedback to help control systems and improve performance. It is one of the most fundamental engineering tools. Re-evaluation should be a continual process with many parallel loops. Reevaluate means observing outputs and using this information to modify the system, the inputs, the product, or the process.

Figure 1 shows the SIMILAR Process. This figure clearly shows the distributed nature of the Re-evaluate function in the feedback loops. However, all of these loops will not always be used. The loops that are used depend on the particular problem to be solved.

This paper is primarily about the Investigate Alternatives function. The four main tasks in investigating alternatives are brain storming to develop the alternative concepts, designing the alternative systems, modeling the systems, and finally performing a tradeoff analysis to find the preferred alternative. The rest of this paper is about tradeoff analyses, which have six subfunctions: create an hierarchical structure, assign weights of importance, select a combining method, design figures of merit, consider the do-nothing alternative, and conduct sensitivity analyses.

3. TRADEOFF ANALYSES

A tradeoff analysis (also called a trade study) is an analytical method for evaluating and comparing system designs based on stakeholder-defined criteria. Tradeoff analyses are most often performed as a part of the Investigate Alternatives function of the SIMILAR process (see http://www.sie.arizona.edu/sysengr/pinewood/ for an example). The cornerstone of performing a tradeoff study is designing Figures of Merit (FoMs). FoMs are also known in the literature as evaluation measures, measures of effectiveness, attributes, performance measures, or metrics. FoMs are quantifiable criteria that are useful in characterizing how the stakeholder values important system attributes. In the context of tradeoff studies, we use FoMs to determine how to trade off preference requirements in order to identify the system design that most closely matches the stakeholder's objectives. The process of designing FoMs assists in the formation of the stakeholder's preference structure through which alternate system designs can be compared. Stakeholder preferences can be described and quantified in a variety of ways. For many years, decision analysts have studied how people make decisions, which has resulted in a multitude of techniques for handling these issues. We will only consider the use of *scoring functions* to characterize, quantify, and provide a common basis for analyzing a stakeholder's preferences with regard to a given FoM. We will go into greater detail about FoMs and scoring functions in Sections 5 and 6, respectively.

3.1. Tradeoff Study Structure

A tradeoff study is commonly structured as hierarchical partitions of FoMs. The hierarchy may be arbitrarily deep; however, Kirkwood [1997] advises that smaller hierarchies are desirable, because they can be communicated more easily, and the analysis requires fewer resources. When actually performing the analysis, each FoM is evaluated through its scoring function and the resulting scores are combined or "rolled up" at each layer of the hierarchy to achieve an overall numerical evaluation of the system under study. The overall numerical evaluation is used to compare system designs at a high level.

In designing the hierarchical FoM structure for a system, we usually group FoMs that have some natural relationship with one another. Sometimes it is beneficial to first list possible categories of concern to the stakeholder and then derive appropriate FoMs that belong to each category identified. Common top-level categories in the hierarchy are performance, cost, schedule, and risk. Each top-level category should be decomposed into finer subcategories as needed to capture the full spectrum of stakeholder criteria.

The hierarchical partitioning of the FoM structure allows us to evaluate contributions from the bottom-level FoMs on up to the top-level categories. The hierarchical structure also helps in dealing with dependence issues between FoMs. Since FoMs in any given category should be intrinsically related, dependence between them remains isolated in that specific category and will not directly contaminate the overall evaluation. See Kirkwood [1997] for additional properties that are desirable in FoM hierarchies.

3.2. Weights of Importance

Another critical element of a tradeoff study is in selecting the weights, through which priorities among FoMs and layers in the FoM hierarchy are established. That is, FoMs or categories with higher importance are given more weight in the overall evaluation of the system. Weights of importance should be assigned keeping in mind the five principles of Von Neumann and Morgenstern [1947] and Howard [1992], namely, that fundamental decision analysis is based on probability, order, equivalence, substitution, and choice. As a first approximation, we typically have the customer assign a number between 1 and 10 and then we normalize the weights in each category. The authors are aware that this method is ad hoc and without axiomatic basis. However, it has been proven to be useful as a rough indicator of stakeholder preferences despite inconsistencies that may occur by employing this method. The important thing is to get the decision-maker to think about these issues. The exact numbers are not that important. After all, your decision-makers are going to be inconsistent, and they will change their minds anyway.

Many other methods for deriving weights exist, including: the ratio method [Edwards, 1977], tradeoff method [Keeney and Raiffa, 1976], swing weights [Kirkwood, 1997], rank-order centroid techniques [Buede, 2000], and paired comparison techniques discussed in Buede [2000] such as the Analytic Hierarchy Process [Saaty, 1980], tradeoffs [Watson and Buede, 1987], balance beam [Watson and Buede, 1987], judgments, and lottery questions [Keeney and Raiffa, 1976]. These methods are more formal, and some have an axiomatic basis. For a comparison of weighting techniques, see Borcherding, Eppel and Winterfeldt [1991].

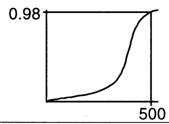
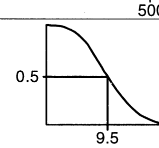
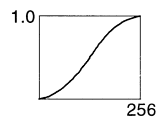
A fragment of a tradeoff analysis showing scoring functions, weights, and a simple combining method is shown in Table II.

4. METHODS FOR COMBINING DATA

In this section, we explore different methods for combining scores, or values in a tradeoff study to calculate a numerical measure that can be used to compare alternatives. The combining methods described here are used to combine data at all levels of the FoM hierarchy.

At the lowest layer when we are dealing with individual FoMs, the scores are given as outputs of the scoring functions associated with each of the FoMs, and the weights are based on expert opinion or customer preference. When we move to higher levels in the hierarchy, the scores are no longer derived from scoring functions, but are derived by combining scores at lower levels. Again, weights at higher layers in the hierarchy

Table II. Tradeoff Study Fragment (Computer System Example)

Figure of Merit							Input value (Score)	Weight	Score times weight
1. **Cost**							0.34	0.50	0.17
2. **Performance**							0.87	0.50	0.435
	Qualitative weight	Normalized weight	Input value	Scoring function	Score	Score times weight			
2.1 **CPU Speed**	10	0.50	500 MHz		0.98	0.49			
2.2 **Hard drive average seek time**	5	0.25	9.5 ms		0.50	0.13			
2.3 **Amount of RAM**	5	0.25	256 Mb		1.0	0.25			
Sum						0.87			
									0.605

are based on expert opinion or customer preference, perhaps from a different category of stakeholder.

Combining values at the top layer of the hierarchy yields the overall numerical designation used to compare alternative systems. A selection of data-combining methods, or tradeoff functions, is outlined next.

4.1. Linear Combination

The linear method of combining data is the simplest and most common method. To describe the data combining process, first suppose there are n reasonably independent FoMs to be combined (perhaps they are in the same layer in the FoM hierarchy). We assign a qualitative weight to each of the n FoMs and then normalize the weights so that they add up to 1. Data are collected for the FoM, each FoM is then evaluated with its scoring function, and the resulting scores (valued from 0 to 1) are then multiplied by the corresponding weight. The final result is the summation of the weight-times-score for each FoM. This process is commonly used, for example, when computing a grade-point average for a student at a university.

The equation defining the process mathematically is given as

$$f = \sum_{i=1}^{n} w_i \cdot x_i ,$$

where n is the total number of FoMs to be combined, w_i represents the normalized weight, and x_i represents the score for the ith FoM. An extensive example of rolling-up of figures of merit is given at http://www.sie.arizona.edu/sysengr/pinewood/.

Linear combination, which is the most common method in engineering, also seems to have great public acceptance, as it is the technique used to rate the National Football League (NFL) professional quarterbacks and also to select the college football teams that will play for the national championship in the Bowl Championship Series (BCS). The National Football League uses these four FoMs:

1. FoM1 = (Completed Passes)/(Pass Attempts)
2. FoM2 = (Passing Yards)/(Pass Attempts)
3. FoM3 = (Touchdown Passes)/(Pass Attempts)
4. FoM4 = Interceptions/(Pass Attempts)

$$\text{Rating} = [5(\text{FoM1} - 0.3) + 0.25(\text{FoM2} - 3) + 20(\text{FoM3}) + 25(-\text{FoM4}+0.095)] * 100/6.$$

Source: November 1998, http://www.sportserver.com/newsroom/sports/fbo/1995/nfl/nfl/stat/qbrating.html, and http://espnet.sportszone.com/editors/nfl/features/qbrate.html.

College football also uses linear combination to schedule a game that might determine the national champion. The Bowl Championship Series (BCS) rating system has been established to determine the teams that will participate in the championship game of the Bowl Championship Series. The ranking system has four figures of merit (called components): subjective *polls* of the writers and coaches, *computer rankings*, *team record*, and *strength of schedule*. The two teams with the lowest total for the four figures of merit play in the national championship game.

Polls: The poll component is calculated based on the average of the ranking of each team in the Associated Press media poll and the ESPN/USA Today coaches poll. The rankings of each team are added and divided by two. For example, a team ranked in No. 1 in one poll and No. 2 in the other poll receives 1.5 points (1 + 2 = 3, 3/2 = 1.5).

Computer rankings: The second component consists of three computer rankings, which are published in major media outlets. These computer rankings include Jeff Sagarin's, published in *USA Today*, the *Seattle Times*, and *The New York Times*. An average of these three rankings is utilized to calculate the points in this component. In order to prevent unusual differences which might occur as the result of individual computer formulas, a maximum adjusted deviation of no greater than 50% of the average of the two lowest (best) computer rankings is utilized in this calculation. For example, if a team is ranked No. 3 and No. 5 in two of the computer systems and No. 12 in the third computer system, the highest ranking of 12 is adjusted to No. 6 before calculating the average points for the computer component (3 + 5 = 8, 8/2 = 4, 4 * 50% = 2, 2 + 4 = 6). The average of these three rankings is calculated for the points of this component (3 + 5 + 6 = 14, 14/3 = 4.67).

Strength of schedule: The third component is the team's strength of schedule. This component is calculated by determining the cumulative won/lost records of the team's opponents and the cumulative won/lost records of the team's opponents' opponents. The formula is weighted two-thirds for the opponent's record and one-third for the opponents' opponents record. The team's schedule strength shall be calculated to determine in which quartile it will rank: 1–25, 26–50, 51–75, 76–100 and is further quantified by its ranking within each quartile (divided by 25). For example, if a team's schedule strength rating is No. 28 in the nation, that team would receive 1.12 points (28/25 = 1.12).

Team record: The final component evaluates the team's won/lost record. Each loss during the season represents one point in this component.

Summary: All four components are added together for the total rating. The team with the lowest point total ranks first in the Bowl Championship Series standings. The BCS standings will be published in the second week of November each season. This system will be utilized only to select the teams that will participate in the championship game of the Bowl Championship Series and to determine any independent team or team from a conference without an automatic selection that shall qualify for a guaranteed selection in one of the games of the Bowl Championship Series as a result of being ranked in the top six in the BCS standings. Source: November 1998, http://espn.sportszone.com/ncf/news/980609/00731091.html.

4.2. Product Combination

The combination function for the product combining method is given as

$$f = \prod_{i=1}^{n} x_i, \qquad \text{or alternatively} \qquad f = \prod_{i=1}^{n} x_i w_i .$$

In the above equations, n represents the number of FoMs that are to be combined, x_i represents the output of the scoring function for the ith FoM, and w_i represents the weight of importance assigned to the ith FoM. The Product Tradeoff Function is commonly used, for example, in computing cost to benefit ratios and in doing risk analyses. This method favors alternatives where all figures of merit have moderate values. It rejects alternatives that have a FoM value of zero. This method is often used for mission critical functions, where setting one function to zero, or a very low number, is not likely to be overcome by extra effort in the other functions.

4.3. Exponential Combination

Cooper [1999] developed the exponential combining method discussed in this section. The application of this exponential technique is effective when incorporating uncertainty into the model.

The exponential model presented here is one example of a nonlinear combining function. The original model incorporates two families of information, one that leads to an increase in preference and another that leads to a decrease in preference. However, in the following model, we only consider data that leads to an increase in preference. The exponential combining function is given as

$$f = 1 - e^{-\sum_{i=1}^{n} k w_i x_i} .$$

The w_i indicate weights (scaled between 0 and 1) that suggest the significance of the ith measure with respect to increases in the overall preference level for the criteria. At the bottom layer of the hierarchy, the x_i represent scores, or output of the scoring function affiliated with the ith FoM. At higher layers, x_i represents the aggregate values from the lower layers. k is a scaling constant used to further tailor the output to match the requirements necessary for accurate evaluation. Note that if $k = 1$, the output of this model will range from 0 to 0.63.

4.4. Sum Minus Product Combination

The sum minus product combining function for two FoMs, x and y, is given as

$$f = x + y - x \cdot y, \quad \text{or alternatively,}$$

$$f = w_1 \cdot x + w_2 \cdot y - w_3 \cdot x \cdot y\,.$$

In this function, x and y are the outputs of the scoring functions for FoMs x and y, respectively. In the alternate form, w_1 and w_2 represent the weights for FoM x and y, respectively, and w_3 represents a weight for the product of the two FoM evaluations. The sum minus product combination function has its origins in probability theory: It is appropriate for computing probabilities of unions for independent events. It also is the function used in Mycin style decision support systems for computing certainty factors when two or more rules with the same conclusion succeed [Buchanan and Shortliffe, 1984].

4.5. Compromise Combination

The compromise combining function for two FoMs, x and y, is given as

$$f = [x^p + y^p]^{1/p}, \quad \text{or alternatively,}$$

$$f = [w_1^p \cdot x^p + w_2^p \cdot y^p]^{1/p}.$$

Here x and y represent the outputs of the scoring functions for FoMs x and y, respectively, and the variable p represents a scaling factor. The Compromise Tradeoff Function identifies alternatives that have the largest distance from the origin [Goicoechea, Hansen, and Duckstein, 1982: Duckstein, Treichel, and El Magnouni, 1993]. If $p = 1$, it is merely the linear combination function mentioned above. This function shows perfect compensation, because a unit reduction in one alternative can be perfectly compensated by a similar increase in another. This type of compensation is typified by optimization techniques such as linear programming. If $p = 2$, it is the Euclidean distance measure, used, for example, to obtain root-mean square measurements. This type of compensation is used in problems with quadratic objective functions, such as certain optimal control problems. If $p = \infty$, then the preferred alternative will be the one that has the largest figure of merit. There is no compensation in this case, because only one of the figures of merit (the largest) is considered. Consider the

$$Compromise\,Output = [x^p + y^p]^{1/p}.$$

If $x > y$ and $p >> 1$, then $x^p >> y^p$ and the

$$Compromise\ Output = [x^p]^{1/p} = x.$$

This type of compensation is typical in H_∞ control or in nuclear reactors where peak centerline temperature has to be limited. Steuer [1986] calls this a Chebyshev norm. The Compromise Tradeoff Function with $p = \infty$ might be most appropriate when choosing a hero and the figures of merit show no compensation, for example, when selecting the greatest athlete of the century using figures of merit based on Number of National Championship Rings and Peak Salary, when picking the baseball player of the week using Home Runs and Pitching Strikeouts, when a couple is choosing a movie using figures of merit based on Romance, Action, and Comedy, when the NBA teams are drafting basketball players using Height and Assists, or when a pregnant woman is choosing between Pickles and Ice Cream.

4.6. Certainty Factors

Certainty Factors (CFs) have been used in expert systems for many years [Buchanan and Shortliffe, 1984]. The underlying method surrounding CFs is based on probability theory and has survived mathematical scrutiny. Thus, a vast knowledge base has developed for CFs, and a great deal is known about their properties and uses.

The initial CF datum point is derived from the weight and score for the first FoM as follows: $CF = w_1 * x_1$, where w_1 is the weight (between 0 and 1) and x_1 is the score (output of the scoring function, also between 0 and 1) corresponding to the 1st FoM. The formula for computing CFs beyond the initial datum point is given in Eq. (1) below:

$$TotalCF = OldCF + (1 - OldCF) * (w_i * x_i) \qquad (1)$$

The CFs for the remaining FoMs are combined using Eq. (1) to create an aggregate score for their respective subcategories. When we move to the next level up, the x_i's and w_i's become the weights and scores for the subcategories just calculated, and so on. At the highest level of the FoM structure, *TotalCF* becomes the overall evaluation from which alternatives can be compared. In our analysis, the CFs are restricted to the range [0, 1].

Another advantage of using CFs as a combining method is that the weights (w_i) do not have to be normalized. This means each time the objectives hierarchy is modified, be it the addition or subtraction of a new FoM, or a layer in the hierarchy, it is not necessary to renormalize the weights as with linear combination. This feature simplifies computations; however, the weights must still be in the range [0, 1].

4.7. Summary

Of course, there are many more methods for combining evidence (see, for example, Goicoechea, Hansen, and Duckstein [1982], Szidarovszky, Gershon, and Duckstein [1986], Edwards [1992], Bardossy and Duckstein [1995], and Buede [2000]). Even listing them is beyond the scope of this paper. We have only mentioned a few methods to give the reader an idea of possible options. In fact, different methods may be used in different parts of the hierarchy. It should be noted that, in some cases, the choice of a combining method is more important than acquiring accurate data about the system [Keefer and Pollock, 1980; Bahill, Dahlberg, and Lowe, 1998]. Therefore, it is at least as important to choose the appropriate combining method as it is in collecting precise data. This is particularly true when the distribution of alternatives in the Figure of Merit space is nonconvex [Bahill, Dahlberg, and Lowe, 1998].

The next section deals in depth with the process of selecting FoMs, or criteria useful in evaluating systems for comparison purposes. The central idea here will be the use of scoring functions. Scoring functions, among other things, make direct comparisons between systems possible. Scoring functions, their uses and properties is the main subject of the rest of the paper.

5. FIGURES OF MERIT

In practice, it is usually desirable to employ a simple, objective, and quantitative method to analyze alternative system designs for comparing attributes such as performance, cost, schedule, and risk. Selecting a technique for conducting these analyses is typically handled with approaches studied in the field of multiattribute utility theory. The general tasks involved in multiattribute utility measurement are structuring objectives and operationalizing attributes, eliciting single-attribute utility functions, eliciting weights, and aggregating weights and utility functions [Borcherding, Eppel, and von Winterfeldt, 1991]. This section of our paper deals with creating figures of merit, which fits into the operationalizing system attributes task.

When evaluating systems and investigating tradeoffs, the analyst and the customer must select measures that encompass the customers' preferences and values regarding system designs with respect to the problem at hand. With these system measures, we can infer about the overall quality or performance of the system as judged by the customer. As mentioned previously in this paper, we call such system measures or properties figures of merit. FoMs are specific items that need to be quantified to determine how well the system under study satisfies the design requirements from the stakeholders point of view [Chapman, Bahill, and Wymore, 1992]. Consider the simplistic example of evaluating a computer system. In comparing computer systems one may use FoMs such as cost, processor speed, amount of RAM, hard disk size, etc., to determine how well each computer system meets the requirements for these metrics. FoMs are most commonly derived for "preference" requirements, or system requirements that show compensation and can be traded off. Sometimes we are faced with the situation where the customer states mandatory requirements in such a way that they are in conflict. It may be necessary to trade off mandatory requirements in these instances.

We shall make a distinction between preference and mandatory system requirements. Mandatory requirements state specific conditions that must hold in order for the system to be considered acceptable. In general, mandatory requirements are stated as "hard" limits and show little or no compensation. In the above example, a mandatory requirement might be that the processor speed must be above 400 MHz. Consequently, any system demonstrating a processor speed below 400 MHz constitutes an infeasible alternative in the eyes of the customer. We will see later that mandatory requirements can play an important role when evaluating FoMs for preference requirements.

5.1. Designing Figures of Merit

5.1.1. Hierarchy

Experience shows that stakeholders are not comfortable with considering more than seven FoMs at a time, and therefore FoMs should be organized in a hierarchical fashion to enable communication of information and systematic aggregation. As we mentioned in Section 3.1, there will usually be an intrinsic partitioning of the FoMs that we should take advantage of by breaking up related FoMs up into smaller, more manageable categories or layers. These natural partitions are combined to form higher-level categories whose value represents the combination of all lower-layer FoMs belonging to the category.

Fundamental or top-layer FoMs are evaluated by aggregating bottom-layer FoMs and all intermediate layers. In the previous computer system example, an intermediate or top-layer FoM such as "computer performance" can be considered a combination of many lower-layer FoMs such as processor speed, quantity of RAM, bus speed, etc., and some of these FoMs should be broken down into even finer granularity as needed. This hierarchical decomposition of the FoM structure is also known as the fundamental objectives hierarchy [Buede, 2000].

5.1.2. Properties of FoMs

When selecting FoMs, combining methods, and verification processes, it is important to ensure that they satisfy certain properties. Some of the properties mentioned here will pertain to the FoMs themselves while some may concern the combining method or verification process for the FoM. However, problems created by violating these properties can usually be ameliorated by reengineering the FoMs.

5.1.2.1. Quantification. FoMs should be *quantitative*, meaning that it must be possible to build a scoring function for the FoM. We desire quantifiable metrics so that we are able to measure and collect data that can be fed directly into the scoring function for evaluation. "Soft" or "fuzzy" metrics such as "safety culture" are not directly measurable as stated, and violate the quantitative property of FoMs. These types of FoMs are good candidates for higher-level categories containing quantifiable FoMs. Alternatively, the analyst should seek out measurable quantities that convey the required information. Customers usually think in terms of "soft" metrics, so it is our job as systems engineers to derive appropriate, measurable metrics that can then be used to objectively evaluate the system with respect to the information sought by the customer. For the "safety culture" FoM stated earlier, one possible measure might be the number of safety training courses administered per year.

5.1.2.2. Independence. FoMs should also be *independent*, implying that information about one FoM should not lead to inferences about another. Dependence causes complications, because in combining the data to extract results we are not obtaining orthogonal or independent combinations. This makes it difficult or impossible to ascertain which of the FoMs is in fact contributing to the analysis and which are confounded due to interactions with other FoMs.

There are four main independence cases when considering systems characterized with multiple FoMs: preferential, weak-difference, utility, and additive independence. See Keeney [1992] for a discussion of these independence cases and for a discussion on how to verify whether or not they hold. Keeney [1992] states that if any of these independence conditions do not hold, it is usually because we have missed some FoMs.

One ad-hoc method used to reduce the effects of dependence is to bundle dependent FoMs into higher-level categories. This way if there are any adverse outcomes due to interdependence, they will remain isolated in that category and will minimize any subsequent contamination of the analysis.

5.1.2.3. Transitivity. The combining method used to aggregate FoM output data should show *transitivity*, meaning if A is preferred to B, and B is preferred to C, then A should be preferred to C. Violations of this property show inconsistencies that can produce invalid conclusions. The fundamental axioms of decision analysis state that this is an essential ingredient to sound analysis. The consequences of infringing on this property can be illustrated by the "money pump" argument [Buede, 2000]; an impartial party may lure us into offering an infinite amount of money by providing us with a sequence of trades among three alternatives. For example, I would be violating this property if I stated that I preferred to live in Albuquerque compared to Berkeley, Berkeley compared to Chicago, and Chicago compared to Albuquerque. With this preference structure and the fact that I live in Albuquerque, I would pay to move to Chicago, pay again to move from Chicago to Berkeley, and then pay a third time to move from Berkeley to Albuquerque where I started out. At this point, it is evident that this preference structure is seriously flawed.

5.1.2.4. Objectivity. FoMs should be *objective* or observer-independent. Data that depend heavily on "engineering judgment" may be more susceptible to bias and error. It has been suggested by Buede [2000] that experimental design, simulation, or similar techniques should be employed to generate data whenever possible to avoid such difficulties. For example, one should avoid such FoMs as "prettiness" or "niceness" for selecting crew members. In sports, most valuable player selections are often controversial. Deriving a consensus for the best football player of the century would be impossible. These examples violate the objectivity property of good FoM design.

5.1.2.5. Statement of FoM. FoMs should be designed and worded so that a higher score results in a more desirable outcome. This aids in interpreting results; i.e., when comparing two alternatives, the alternative with the higher score should be preferred to the other. If scoring functions are used, the shape of the scoring function will determine how scores are allocated; this allows more flexibility in the statement of the FoM. However, the wording of the FoM should still be considered with care; i.e., the use of double negatives or complex phrasing should be avoided. If scoring functions are not used, then wording the FoM such that a higher score is desirable will satisfy this property.

5.1.2.6. Temporal Order. The *temporal order* of verifying or combining FoMs should not be important; e.g., the clothing for our astronauts should be Flame Proof and Water Resistant and verification should not depend on which we test first. Questionnaires are often written to lead the reader down a primrose path. Whether a question is asked early or late in a list might effect the given answer. Let us consider a combining function (CF) that adds two numbers truncating the fraction: (0.2 CF 0.6) CF 0.9 = 0; however, (0.9 CF 0.6) CF 0.2 = 1. The answer depends on the order in which the inputs were presented, and therefore it would not be a good combining function for a tradeoff study. In the field of Boolean Algebra, it is said that such functions do not obey the Law of Association, e.g., as with the Boolean NAND function (↑): (0↑1) ↑1 = 0; however, (1↑1) ↑0 = 1. If temporal order is important, then an expert system might be more appropriate than a tradeoff study.

5.1.2.7. Compensation. FoMs for preference requirements should show *compensation* meaning they can be traded off. For an example of perfect compensation, consider astronauts growing food on a trip to Mars. Assume we have two FoMs: amount of rice grown and amount of beans grown with the single objective of maximizing yield. A lot of rice and a few beans is preferred equally to a lot of beans and little rice. This shows that we are able to trade off beans for rice and vice-versa. Now consider another scenario that demonstrates no compensation: a system that produces oxygen and water for our astronauts. A system that produced a huge amount of water, but no oxygen might get the highest score, but clearly, it would not support life for long. Care must be exercised in stating tradeoffs between FoMs for requirements that do not show compensation to reveal situations as described above. FoMs of this type often fall into the mandatory requirements class.

6. SCORING FUNCTIONS

A tradeoff study is carried out by assessing each FoM and combining the resulting data (usually on multiple levels) to give a general numerical analysis of the system. For complex systems with many FoMs, it may be difficult or impossible to directly evaluate each FoM in raw units and combine them to elicit useful data. This difficulty is due to the lack of a common basis for combination and comparison (i.e., comparing hard disk space in gigabytes to processor speed in megahertz). Standard scoring functions provide a simple technique through which this problem can be remedied. This method is similar to utility, value or fuzzy-set functions.

Each bottom-layer FoM is assigned a scoring function. Intermediate or top-layer FoMs may also be assigned scoring functions as prescribed by the stakeholder, although Buede [2000] and Wymore [1993] advise that scoring functions are not necessary beyond the bottom layer. The scoring function itself is a mathematical mapping between the FoMs "natural" or measured values to a "coded" or normalized range of values common to all FoMs. It is formalized as a mathematical function accepting parameters specific to the FoM and assigning as output a real number, commonly between 0 and 1, but can be any other consistent scaling. With this technique, a FoM, no matter what raw units or scale it may be defined in, can be transformed by its scoring function to have a set of possible values common to all other FoMs whose raw values have been transformed by their own scoring function. Through this mathematical transformation, a common basis for aggregation and comparison can be attained.

The classical procedure for creating a scoring (or utility) function for a given FoM is to survey the decision maker concerning his or her preferences and judgments as the input values of the FoM are considered [Winston, 1994]. This information guides the researcher in defining the shape of the scoring function curve. The shape of the curve is important as it explicitly illustrates the degree of variation in the customer's judgment with each incremental change in FoM input values. With the classical methods, it is essential that sufficient data be obtained so that the shape of the curve reflects an accurate portrayal of the customer's assessments. As one might imagine, this method can be tedious and time-consuming. To speed up this process, a finite set of "standard" scoring functions can be developed that possess the properties previously mentioned. Standard scoring functions are characterized by a few parameters and enjoy the ability to take on many shapes. This would allow the researcher to conveniently capture a customer's preferences by only specifying a few parameters for a predefined curve.

6.1. Standard Scoring Functions

The construction of standard scoring functions is set up such that the values along the y-axis represent the output indicating customer "happiness" on a 0-to-1 or equivalent scale, and the information associated with the x-axis correspond to the units and scale of measure for the FoM. For example, if we are considering computer processor speed as a FoM, suitable units of measure along the x-axis might be gigahertz. The *range* of values spanning the x-axis is more difficult to establish and will be discussed in more detail later. By output, we are indicating values along the y-axis that are dependent on inputs, represented by values specified along the x-axis.

6.1.1. Exponential Family Scoring Functions

Kirkwood and Sarin [1980] studied the exponential scoring function form and showed that, in many cases, the shapes attained with this form are reasonable. Buede [2000] shows that the exponential form takes on four general shapes: decreasing returns to scale (RTS), linear RTS, increasing RTS, and an S-curve (see Fig. 2). To generate the scoring functions shown in Figure 2, it may be necessary to invert and/or splice portions of the exponential curves. Returns to scale indicates what happens to the output when the input is successively increased by the same proportion.

The general form used to generate the curves is an exponential function that accepts three parameters and an input value. The specific exponential forms are shown as Eqs. (2a) and (2b) as presented by Kirkwood [1997]. Equation (2a) represents preferences that are monotonically increasing over the input value v. This implies that the *higher* the value of v, the higher the *Score*, or the more preferred it is. Equation (2b) represents a monotonically decreasing scoring function over the input value v. With the monotonically decreasing curve, the *lower* the value of v, the higher the *Score*, or the more preferred it is. As exhibited by Kirkwood [1997], the indices representing individual input values and FoMs are left off to simplify the presentation of the equations.

$$Score = \begin{cases} \dfrac{1 - e^{-(v - Low)/\rho}}{1 - e^{-(High - Low)/\rho}}, & \rho \neq \text{infinity}, \\ \dfrac{v - Low}{High - Low}, & \text{otherwise}, \end{cases} \tag{2a}$$

$$Score = \begin{cases} \dfrac{1 - e^{-(High - v)/\rho}}{1 - e^{-(High - Low)/\rho}}, & \rho \neq \text{infinity}, \\ \dfrac{High - v}{High - Low}, & \text{otherwise}. \end{cases} \tag{2b}$$

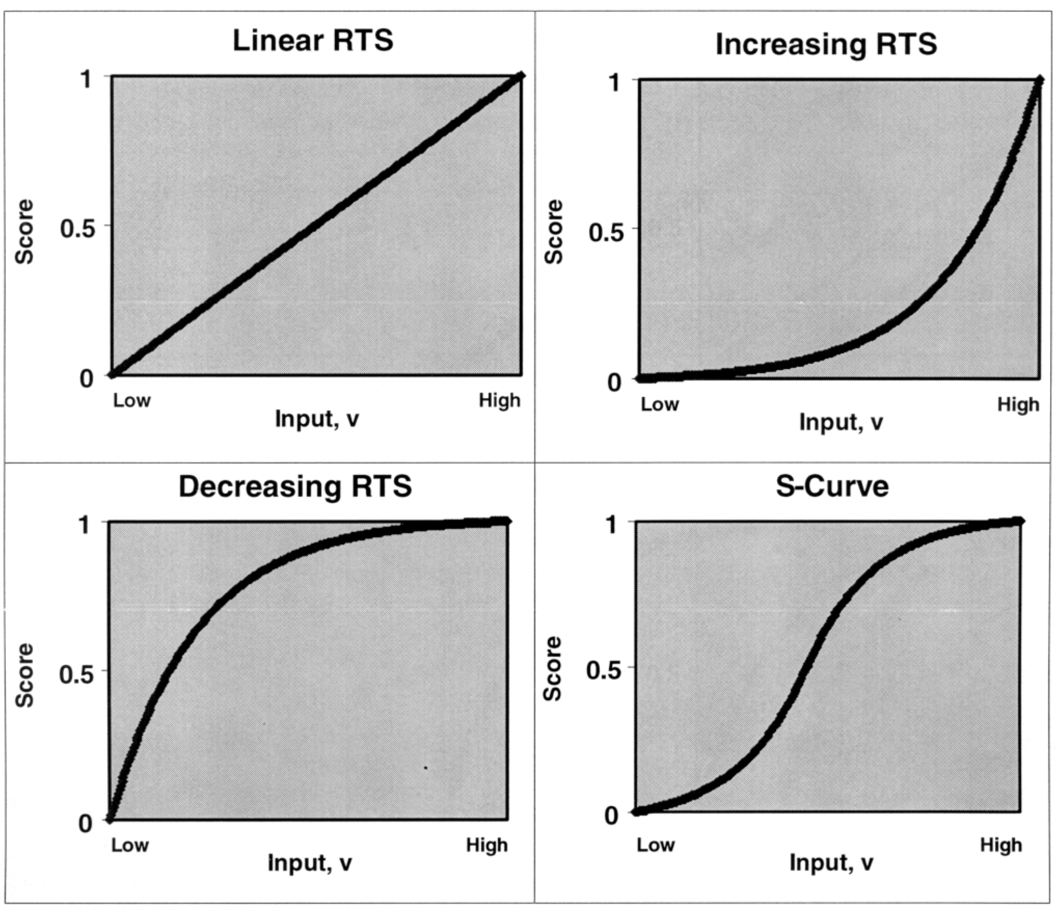


Figure 2. Four basic exponential scoring function forms.

In both Eqs. (2a) and (2b), *Score* represents the output of the function corresponding to the input value v. Note that *Score* will always be between 0 and 1.

The parameters *Low* and *High* define the lower and upper bounds, respectively, of the domain for the FoM. The domain encompasses inputs v to the FoM that can be demonstrated by the alternatives under consideration. The domain bounds may be established due to buildability (budget or technological) constraints or due to mandatory requirements (see Section 5: Figures of Merit). If the bounds on the domain are established by mandatory requirements, alternatives exhibiting values outside the domain are eliminated from consideration, or seriously reconsidered.

Buede [2000] in addition identifies a range of input values v called the *key performance range* that encapsulates the FoM input values that are of most interest to the stakeholder. In this range, the rate of change of the stakeholder's preferences with respect to the input data is at or near the maximum. Input values greater than or less than the key performance range thresholds result in a diminishing rate of change in customer preference. That is, we begin to care less and less about values outside of this key range.

The intersection of the key performance range and the domain truncates the curve such that the key performance range translates the curve along the x-axis, while the domain effectively "cuts off" values outside of its bounds. The exponential scoring functions are conceptually constructed by first identifying the upper and lower limits (*Low* and *High*) of the domain and then observing how the customer's preferences (key performance range) overlap these values. In other words, the domain is defined in terms of what can physically be realized from a buildability or requirements standpoint, and then the curve is constructed based on the customer's preferences in this region. Figures 3–6 show the four basic scoring functions that can be derived with this exponential function by splicing together two exponentials and then perhaps truncating one end or the other. Next, we will give general guidelines for applying these curves and we give a short example for each.

Figure 3 shows a Decreasing RTS function.

Guideline: With decreasing returns to scale (returns to scale indicates what happens to the output when the input is successively increased by the same proportion), we care equally for input data at the lower and middle range of the domain, but there exists an input point at which we care less and less about inputs increasing past this value. There is an implicit assumption that more is always better.

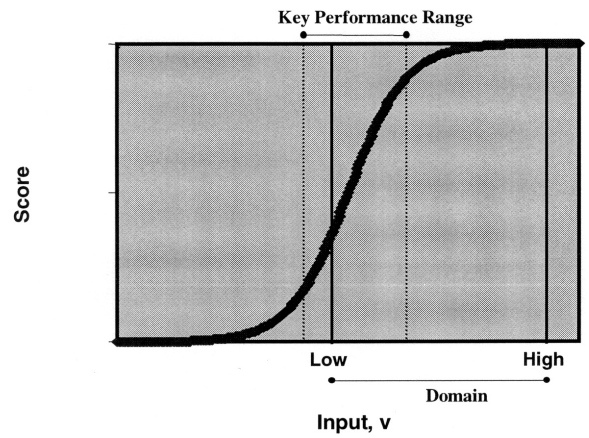


Figure 3. Decreasing RTS. Key performance range (KPR) exceeds Low.

Example: Let the units be the number of trips to an all-you-can-eat salad bar. At first, each trip is valued equally; however, after a number of trips the value one perceives by each incremental trip diminishes, even though we still consider more as better.

Figure 4 shows an Increasing RTS function.

Guideline: Use increasing returns to scale if the domain specifies an input value at which the stakeholder begins to care less and less about change *below* this value. At the other end, increases are valued more or equally well up to the upper limit of the domain. At the upper end, we have not reached a point where the change in the customer's values diminishes past that input.

Example: Consider evaluating a manufacturing process based on a Throughput FoM. Assume

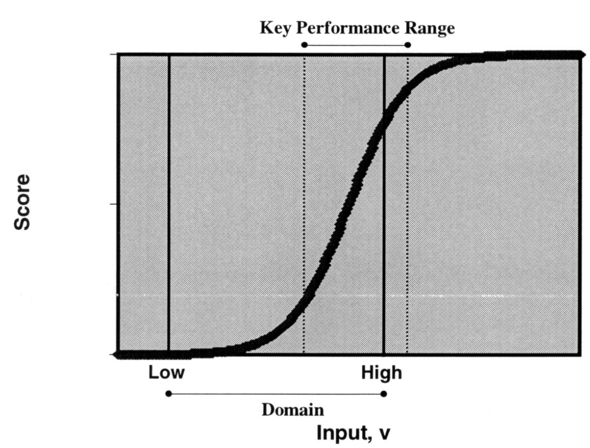


Figure 4. Increasing RTS. KPR exceeds High.

that the design alternatives define a domain of possible input values as low as 100 widgets per hour and technology constraints limit the number of widgets per hour to 200. If the current competitor is able to produce at 190 widgets per hour, inputs below say 170 will be of limited value and inputs around 170 will be considered equally or more valued up to the upper technological threshold (and past it if that were legal).

Figure 5 shows a Linear RTS function.

Guideline: Linear returns to scale occur when the domain restricts the possible input values so that, throughout the domain, changes in those inputs are valued equally. There is no input point in the domain range at which changes past that point cause the stakeholder to care less and less about that change. More is always considered equally better, and less is always considered equally worse.

Example: Consider any situation where some constraints such as technology, budget, etc. limit the valid input data in the range where the customer most values changes in the input. One example of a linear tradeoff is in currency exchange. The exchange rate is constant irrespective of how much currency is to be exchanged.

Figure 6 shows an S-curve.

Guideline: Use an S-curve if the domain permits input values that are greater than and less than the key performance range as defined by the customer. This means that the design alternatives being considered allow input values for the FoM that more than satisfy the customer's preferences at the upper end, and allow input values at the lower end that the customer sees a diminishing rate of change in preference. "The S-curve indicates that the range of possible designs has been maximized" [Buede, 2000, p. 363].

Example: Consider the widget-manufacturing scenario in the above example. If the domain of possible values now allows a lower bound of 100 and an upper bound of 5000 widgets produced per hour, we will still care about increases above the benchmark of 190, but only up to a certain point past which we will not value increases as much anymore. Similarly, values below the 190 benchmark begin not to interest us and thus will still be valued less, but at a decreasing rate. Inputs between these two ranges are assigned the most change in value per unit change in input.

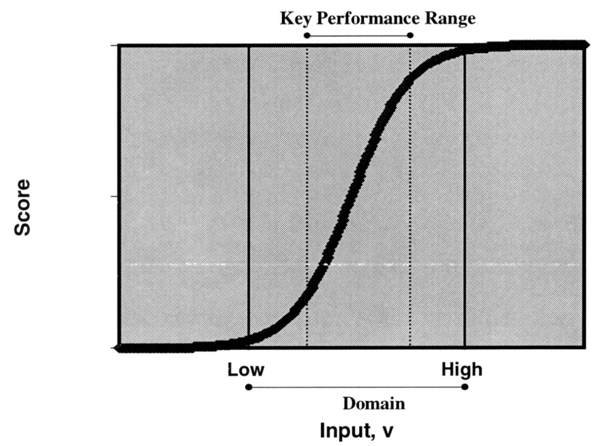


Figure 6. S-curve. KPR is within Low to High.

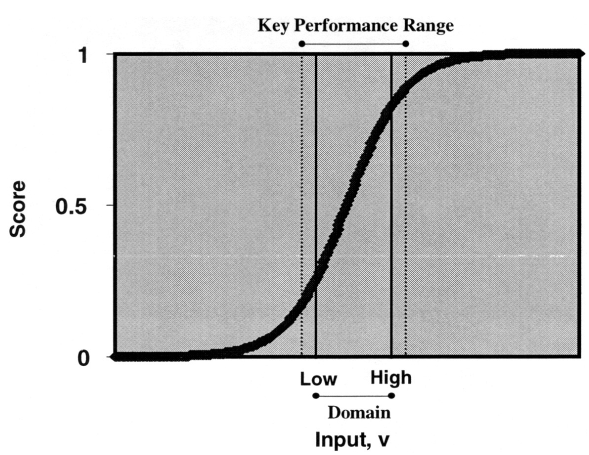


Figure 5. Linear RTS. KPR exceeds Low and High.

The value assigned to the parameter ρ, the exponential constant in Eq. (2), defines the shape of the curve. According to Kirkwood [1997], the appropriate values for the parameter ρ will depend on the range of the domain (*High* – *Low*). Realistic values of ρ will typically have a magnitude greater than one-tenth of the domain range. There is no upper limit to the value of ρ, but if the magnitude is greater than ten times the domain range, the curve will almost be a straight line. In this case, we would just use a straight line. For a more detailed, systematic discussion of choosing a value for ρ (see Kirkwood [1997]).

Additional scoring functions can be derived by combining curves and/or flipping the curves along the *y*-axis. The possible output values for each scoring function shown consistently range from 0 to 1.

6.1.2. Wymore's Scoring Functions

Wymore [1993] has also developed a set of standard scoring functions. In this set are 12 families of scoring functions representing 12 fundamental shapes. A list describing the general shape of each scoring function is given next. Note that these functions limit their output values to a 0-to-1 scale.

SSF1: a shape whose output is 0 as the input goes from $-\infty$ to a point L and then increases asymptotically to 1;

SSF2: a shape whose output is asymptotic to 0 as the input approaches $-\infty$, as the input increases the output increases to 1 at a point U and gives an output of 1 from there on;

SSF3: a shape whose output is 0 as the input goes from $-\infty$ to a point L and then increases to 1 at a point U and gives an output of 1 from that point to ∞;

SSF4: a shape whose output is asymptotic to 0 as the input approaches $-\infty$ gives an output that is asymptotic to 1 as the input approaches ∞ and is always increasing;

SSF5: a hill shape whose output is 0 until the input reaches a point L, then increases to 1 at input point O, then gives an output that decreases to 0 at a point U and is 0 for larger input values;

SSF6: a hill shape whose output is asymptotic to 0 at both ends, but increases to 1 at an input point O.

The notation SSFx refers to Standard Scoring Function x. The other six shapes are obtained by interchanging the roles of 0 and 1 and by replacing "increasing" with "decreasing." Figure 7 shows the six fundamental shapes of the Wymorian standard scoring functions.

Each of the 12 scoring functions is defined mathematically in terms of the first scoring function called SSF1. This is convenient since if any of the 12 scoring functions needs to be evaluated, only one calculation involving SSF1 needs to be done. Figure 8 shows the SSF1 shape along with its mathematical equation. Definition of the parameters:

v: The input value for the FoM.

Score: The output of the scoring function.

L: The lower threshold of performance for the FoM below which the value to the customer is undesirable (but not necessarily unacceptable) and is assigned a zero score.

B: The parameter B is called the baseline value for the FoM and can be chosen as the design goal or the status quo for this or similar systems. By *definition*, baseline values are always assigned a score of 0.5.

S: The parameter S determines the behavior of the scoring function in the neighborhood of the baseline value B. Mathematically, S is the slope of the tangent to the scoring function at the baseline value B. Practically speaking, the slope represents the maximum incremental change in the customer's quantitative judgment with each incremental change in input.

D: The parameter D represents the domain of definition of the scoring function. This is the same concept that was discussed earlier for the exponential family scoring functions in that D clearly states the range of input values that are possible from a buildability viewpoint or legal due to mandatory requirements. Values outside this range constitute impossible or unacceptable inputs.

For the nonmonotonic "hill" and "valley" shaped scoring functions (SSF5, SSF6, SSF11, and SSF 12), we must define additional parameters such as multiple baseline (*B1* and *B2*) and slope (*S1* and *S2*) values, as well as an optimum (O) or pessimum (P) value. The optimum and pessimum values indicate the peak or minima, respectively, of the customer's preferences relative to the input values. The multiple baseline and slope values have the same interpretation as the single parametered counterparts; it just depends on how and where the customer's preferences change before and after the optimum or pessimum values.

One observation about the 12 fundamental Wymorian shapes is that there is always a flattening out of the curves at the extreme input values for a given scoring function. This occurs because, in contrast to the exponential scoring functions whose input range is bounded in terms of what can be built, Wymore's curves are characterized in terms of the customer's preferences. Wymore's scoring functions are constructed by first querying the customer for parameters describing his or her preferences. These parameters or thresholds should contain the key performance range (linear and near-linear region of the curve), and sufficient breadth so that a diminishing rate of change in preference is observed at both the upper and lower limits. This accounts for the flattening out at the extreme points.

However, the full range of input values delimited by the customer's preferences may not be attainable from a buildability or requirements perspective. With Wymorian scoring functions, we allow the domain to implicitly truncate the curve at these hard limits. The logic and reasoning behind any such truncation should be clearly explained in the documentation and denoted on the curves themselves (see Fig. 9).

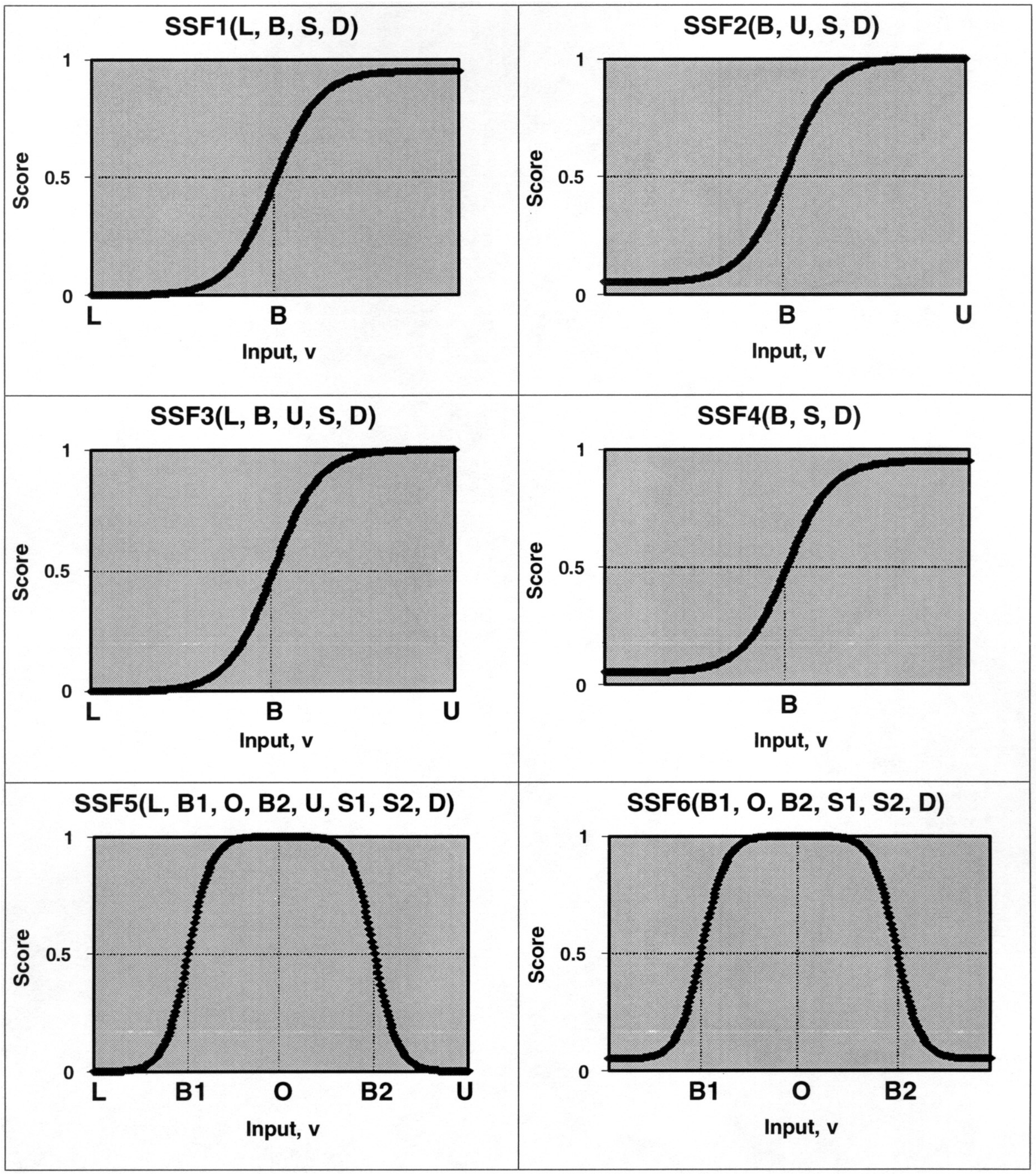


Figure 7. Six of Wymore's standard scoring functions.

We will now present guidelines for choosing an appropriate Wymorian scoring function along with a short example for each (see also Table III):

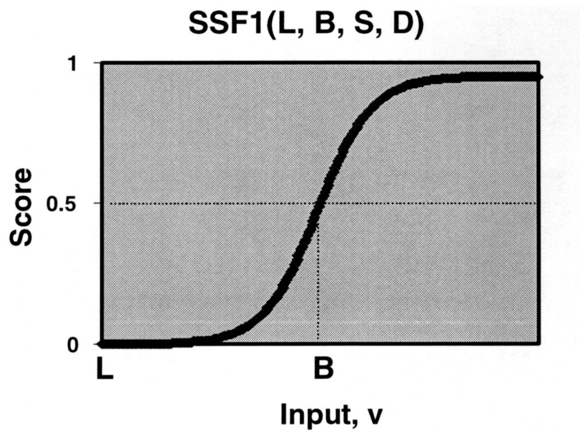
SSF1 Guideline: If more is better, the customer can provide a finite lower bound but no finite upper bound, and the domain can support this, then choose SSF1.

Example: Let the units be the capable density in dots per inch of a printer. The higher the better, and there is no finite upper bound to this measure.

SSF2 Guideline: If more is better, the customer can provide a finite upper bound but no finite lower bound, and the domain can support this, then choose SSF2.

Example: Depth below the surface of the earth, with greater depth being better. There is a fixed upper limit (3951 miles), but as we go out into space (the left side of our function) the value goes to $-\infty$.

SSF3 Guideline: If more is better, the customer can provide both a finite upper bound and a finite



L is the lower threshold
B is the baseline value
S is the slope of the tangent to the curve at B
v is the value of the figure of merit
If v < L then Score = 0
If v ≥ L then Score =

$$\frac{1}{1+\left(\dfrac{B-L}{v-L}\right)^{2*S(B+v-2*L)}}$$

Figure 8. The first Wymorian standard scoring function.

lower bound, then choose SSF3. This is by far the most common scoring function used when dealing with an FoM where more is better.
Example: Let the units be the number of widgets sold. The customer could provide a lower bound

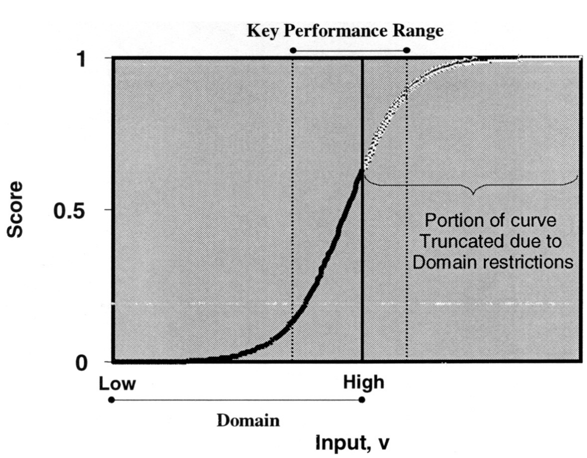


Figure 9. Scoring function truncated by the domain.

below which we can assign a 0 score and an upper bound where anything above that value is assigned a score of 1.

SSF4 Guideline: If more is better, the customer can provide neither a finite upper bound nor a finite lower bound on his or her preferences, and the domain can support this, then choose SSF4.
Example: Let the upper range denote happiness and the lower range represent unhappiness. There are no upper or lower bounds, but more happiness is better.

SSF5 Guideline: If the customer's preferences can be described as a hill-shaped curve where there is one optimum point and values to the left and right of this point are less desirable, and the customer can provide a finite upper bound, a finite lower bound and the optimum value, then choose SSF5.
Example: Let the units be a measurement specification for a particular piece of material. The optimum would be the target value and values greater or less than the target value would result in a lower score.

SSF6 Guideline: If the customer's preferences can be described as a hill-shaped curve where there is one optimum point and values to the left and right of this point are less desirable, and the customer can furnish the optimum value, but not finite upper or lower bounds, then choose SSF6.
Example: Let the units be the error in measurement for a particular quantity and the customer cannot say how low or how high the error would have to be in order to assign a score of 0 or 1. The optimum value would obviously be 0 error.

SSF7 Guideline: If more is worse, and the stakeholder can provide a finite lower bound but no finite upper bound (as in the case of a cost), then choose SSF7.
Example: Number of deaths in airplane crashes. The lower bound would be 0 deaths and we assume the customer cannot give an upper limit where if the number of deaths is greater than this limit it is assigned a 0 score. However, the domain may restrict this use of the scoring function, because the population of the world is finite.

SSF8 Guideline: If more is worse, and the stakeholder can provide a finite upper bound but no finite lower bound, then choose SSF8.
Example: Let the units be the logarithm of the size of a RAM cell. The lower the better, but it may be the case that the customer cannot specify how small the cell can possibly be in order to receive a score of 1. Values approach 0 as the input approaches the upper bound.

Table III. The 12 Wymorian Scoring Functions

Scoring Function	Guideline for Using the Scoring Function
SSF1	**Guideline:** If more is better, the customer can provide a finite lower bound but no finite upper bound, and the domain can support this, then choose SSF1. **Example:** Let the units be the capable density in dots per inch of a printer. The higher the better and there is no finite upper bound to this measure.
SSF2	**Guideline:** If more is better, the customer can provide a finite upper bound but no finite lower bound, and the domain can support this, then choose SSF2. **Example:** Depth below the surface of the earth, with greater depth being better. There is a fixed upper limit (3951 miles), but as we go out into space (the left side of our function) the value goes to $-\infty$.
SSF3	**Guideline:** If more is better, the customer can provide both a finite upper bound and a finite lower bound, then choose SSF3. This is by far the most common scoring function for FoMs where more is better. **Example:** Let the units be the number of widgets sold. The customer can provide a lower bound below which we can assign a zero score and an upper bound where anything above that value is assigned a score of one.
SSF4	**Guideline**: If more is better, the customer can provide neither a finite upper bound nor a finite lower bound on his or her preferences, and the domain can support this, then choose SSF4. **Example**: Let the upper range denote happiness and the lower range represent unhappiness. There are no upper or lower bounds, but more happiness is better.
SSF5	**Guideline**: If the customer's preferences can be described as a hill-shaped curve where there is one optimum point and values to the left and right of this point are less desirable, and the customer can provide a finite upper bound, a finite lower bound and the optimum value, then choose SSF5. **Example**: Let the units be a measurement specification for a particular piece of material. The optimum would be the target value and values greater or less than the target value would result in a lower score.
SSF6	**Guideline**: If the customer's preferences can be described as a hill-shaped curve where there is one optimum point and values to the left and right of this point are less desirable, and the customer can furnish the optimum value, but not finite upper or lower bounds, then choose SSF6. **Example**: Let the units be the error in measurement for a particular quantity and the customer cannot say how low or how high the error would have to be in order to assign a score of zero or one. The optimum value would obviously be zero error.
SSF7	**Guideline**: If more is worse, and the stakeholder can provide a finite lower bound but no finite upper bound (as in the case of a cost), then choose SSF7. **Example**: Number of deaths in airplane crashes. The lower bound would be zero deaths and we assume the customer cannot specify an upper limit where more deaths are not worse. However, the domain may restrict this use of the scoring function, because the population of the world is finite.
SSF8	**Guideline**: If more is worse, and the stakeholder can provide a finite upper bound but no finite lower bound, then choose SSF8. **Example**: Let the abscissa be the logarithm of the size of a RAM cell. The smaller the better, but the customer might not be able to specify how small the cell should be in order to receive a score of one.
SSF9	**Guideline**: If more is worse, and the stakeholder can provide both a finite upper bound and a finite lower bound, then choose SSF9. This scoring function is the most common for FoMs where more is worse. **Example**: Costs. It is usually the case that the customer can provide upper and lower limits on expenditures. For simplicity, we assume the higher the expenditure, the lower the score.
SSF10	**Guideline**: If more is worse, the stakeholder can provide neither a finite upper bound nor a finite lower bound, and the domain can support this, then choose SSF10. **Example**: Assume the customer is trying to build the most powerful microscope. Let the units be the size of the smallest viewable object. Here, the smaller the object, the happier the customer.
SSF11	**Guideline**: If the stakeholders' preferences can be characterized by a valley-shaped curve, there is one pessimum value where inputs greater or less that this value give more desirable results, and the stakeholder can provide a finite upper bound, a finite lower bound and a pessimum value, then choose SSF11. **Example**: Consider a vehicle with unbalanced tires. At a certain speed the car will *shimmy*, which creates an undesirable condition. If the driver maintains speeds far away (either faster or slower) from this pessimum speed, then the result is a smoother, more desirable ride.
SSF12	**Guideline**: If the stakeholders' preferences can be characterized by a valley-shaped curve, there is a pessimum value with inputs greater or less that this value giving more desirable results, and the stakeholder can provide the pessimum value but can provide neither a finite upper bound nor a finite lower bound, then choose SSF12. **Example**: Consider trying to eliminate bacteria growth by varying temperature. There will be a most undesirable temperature where the bacteria grow very fast, and temperatures greater or less than this value will impede bacteria growth.

SSF9 Guideline: If more is worse, and the stakeholder can provide both a finite upper bound and a finite lower bound, then choose SSF9. This scoring function is by far the most frequently used with FoMs where more is worse.

Example: Costs. It is usually the case that the customer can provide upper and lower limits on expenditures. For simplicity, we assume the higher the expenditure, the lower the score.

SSF10 Guideline: If more is worse, the stakeholder can provide neither a finite upper bound nor a finite lower bound, and the domain can support this, then choose SSF10.

Example: Assume the customer is trying to build the most powerful microscope. Let the units be the size of the smallest viewable object. Here, the smaller the object, the happier the customer.

SSF11 Guideline: If the stakeholder's preferences can be characterized by a valley-shaped curve, there is one pessimum value where inputs greater or less that this value give more desirable results, and the stakeholder can provide a finite upper bound, a finite lower bound and the pessimum value, then choose SSF11.

Example: Consider a vehicle driving with unbalanced tires. It may be the case that if the driver reaches a certain speed the car will begin to "shimmy," which creates an undesirable condition. If the driver maintains speeds further and further away (either faster or slower) from this pessimum speed, then the result is a smoother, more desirable ride.

SSF12 Guideline: If the stakeholder's preferences can be characterized by a valley-shaped curve, there is one pessimum value where inputs greater or less that this value give more desirable results, and the stakeholder can provide the pessimum value but can provide neither a finite upper bound nor a finite lower bound, then choose SSF12.

Example: Consider trying to eliminate bacterial growth by varying temperature. There will be a most undesirable temperature where the bacteria grow very fast, and temperatures greater or less than this value will impede bacterial growth.

We mentioned earlier that all 12 fundamental Wymorian scoring functions are defined in terms of SSF1. Next, we will expand this notion by illustrating the explicit manner in which the four basic scoring functions are constructed. We will start with SSF3.

SSF3 has the following four ranges:

1. For input values of the FoM less than the lower threshold L, *Score* = 0.
2. For input values of the FoM between the lower threshold L and the baseline value B, *Score* = *SSF1*(L, B, S) evaluated at v = input value, or *Score* = *SSF1*(L, B, S)(*input value*) with domain D.
3. For input values of the FoM between the baseline value B and the upper threshold U, *Score* = 1 – *SSF1*($2 * B - U$, B, S) evaluated at $v = (2 * B -$ *input value*), or *Score* = 1 – *SSF1*($2 * B - U$, B, S)($2 * B -$ *input value*) with domain D.
4. For input values of the FoM greater than the upper threshold U, *Score* = 1.

SSF9 has similar ranges as SSF3, but is constructed by taking 1 – *SSF3*(L, B, U, $-S$, D) and evaluating at v = *input value*, or *Score* =1 – *SSF3*(L, B, U, $-S$, D)(*input value*). For example, say we have an SSF9 scoring function for a FoM defined with the following parameters: $L = 1$, $B = 2$, $U = 3$, $S = -4$, D = real numbers between 0 and 10, and say the input value v is 2.7. First, note that 2.7 is a valid input since it is within the domain D. Then evaluate the score (denoted "Score" below) for SSF9 through a series of calculations as follows:

$$Score = 1 - SSF3(1, 2, 3, -(-4))(2.7),$$

where

$$SSF3(1, 2, 3, -(-4))(2.7) =$$

$$1 - SSF1(2 * 2 - 3, 2, -(-4))(2 * 2 - 2.7)$$

since the input value, 2.7, is between the baseline value, 2, and the upper limit, 3, where

$$SSF1(2 * 2 - 3, 2, -(-4))(2 * 2 - 2.7) =$$

$$\frac{1}{1+\left(\dfrac{2-(2*2-3)}{(2*2-2.7)-(2*2-3)}\right)^{2*(-(-4))(2+(2*2-2.7)-2*(2*2-3))}}.$$

We have left the numerical values unsimplified so that it can be seen how each value traces back to the original SSF9 problem.

The nonmonotonic Wymorian scoring functions (hill- and valley-shaped curves) are constructed from the appropriate monotonic functions. The upper limit for the first curve must match the lower limit for the second to obtain a smooth transition from curve to curve.

Nonmonotonic scoring functions such as hill and valley shapes are often used when we are trying to balance two or more objectives, or anticipating the actions of others [Keeney, 1992].

6.1.2.1. Mathematical Properties of the Wymorian Scoring Functions. Wymore's base scoring function SSF1 was chosen in part because it has many desirable mathematical properties. Among other attractive results, these properties make it possible to conduct analytical sensitivity analyses when evaluating alternative systems. We will now abbreviate *SSF1*(L, B, S, D) by f1 where f1(v) denotes *SSF1*(L, B, S, D)(v), or SSF1 evaluated at v given L, B, S, and D. The properties are as follows (taken from Wymore [1993: 389–390]:

1. The output of f1 evaluated at the baseline value B is 0.5.
2. The limit of f1(v) as v approaches infinity is 1.
3. The limit of f1(v) as v approaches L from the right is 0.
4. Under the assumption that the slope S is strictly less than $1/(B - L)$:
   a. The limit of the derivative of f1(v) with respect to v as v approaches L from the right is 0.
   b. The value of the derivative of f1(v) with respect to v is positive for every v that is greater than L, and therefore f1 is strictly increasing over the interval (L, ∞)
   c. The value of the derivative of f1(v) with respect to v at the point $v = B$ is S.
   d. The value of the second derivative of f1(v) with respect to v at the point $v = B$ is 0.
   e. The limit of the second derivative of f1(v) with respect to v as v approaches L from the right is 0.
5. From these facts and the form of the function f1, it can be concluded that the first and second derivatives of f1(v) with respect to v are continuous everywhere in D.

Because none of the mathematical manipulations performed on f1 to create the other 11 fundamental scoring functions affect the derivatives, all the Wymorian standard scoring functions as well as their first and second derivatives are continuous.

6.1.2.2. How Many of the Wymorian Scoring Functions Are Necessary? Over the last three decades of using scoring functions (see, e.g., Chapman, Bahill,

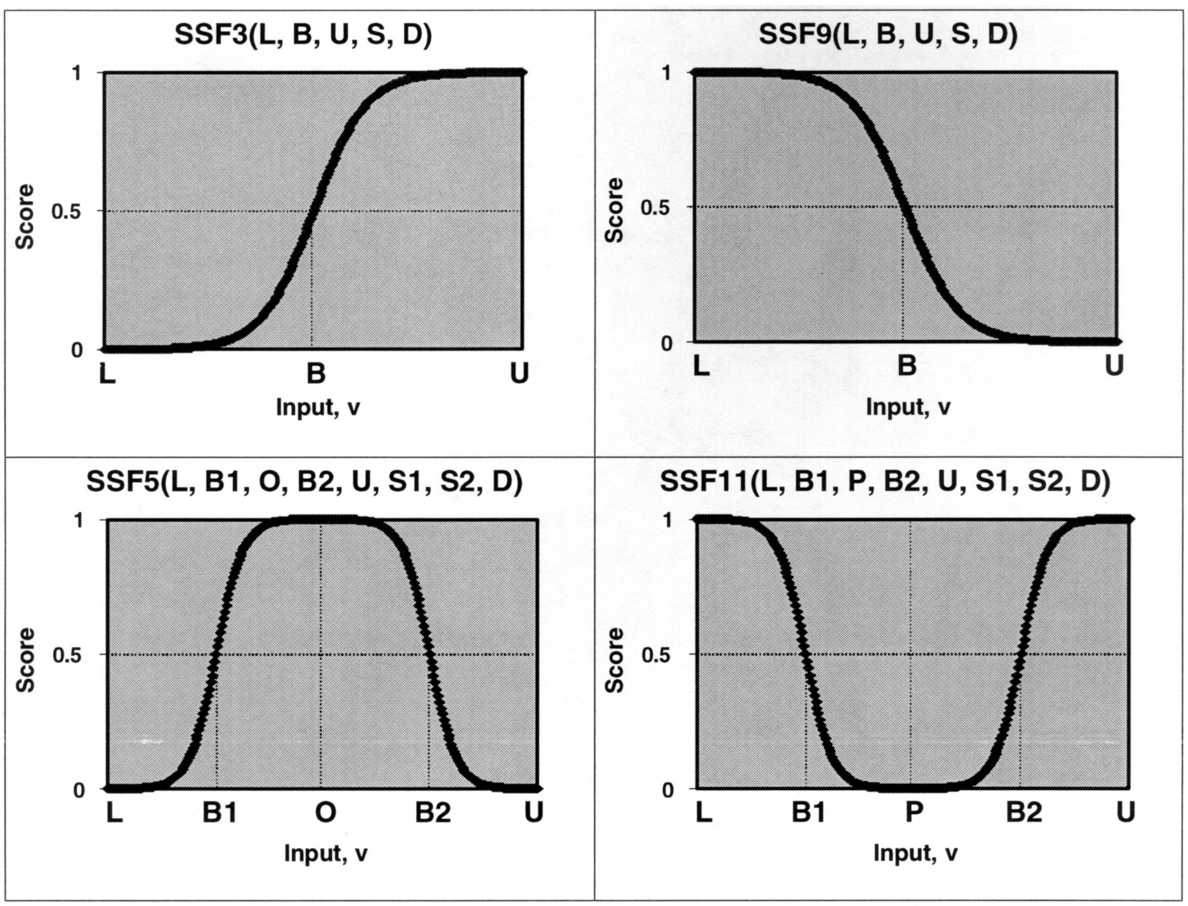


Figure 10. The four essential Wymorian standard scoring functions.

and Wymore, [1992] and http://www.sie.arizona.edu/sysengr/pinewood), we have found that eight standard scoring functions are sufficient for all applications that we have encountered. We use two Boolean or step functions, two linear functions, and the four function shapes shown in Figure 10. The step functions are used when a yes/no type input response is required and there is no continuum of possible inputs. We have also found that infinity end points (as with SSF1) are not necessary. It is adequate to use a large number and adjust the slope accordingly. Of these eight standard scoring functions, we have found the linear scoring functions to be the least frequently used. A program that implements the four Wymorian scoring functions of Figure 10 is available at http://www.sie.arizona.edu/sysengr/slides/.

6.1.3. The Do Nothing Alternative

In all tradeoff studies, it is essential to include a Do Nothing alternative. There are two interpretations of the Do Nothing alternative: (1) Evaluate the status quo. (2) Do absolutely nothing. Bahill owns a 1971 Datsun 240Z that has 150,000 miles on it. He might want to select a replacement for it. His alternatives would be a Honda S2000, a Saturn, and Do Nothing. The two interpretations for Do Nothing are (1) the status quo, keep the 240Z and (2) do without a car, i.e., walk or take the bus. The most common of these approaches is to use the status quo as the Do Nothing alternative.

If the status quo Do Nothing alternative attains one of the highest scores in the tradeoff analysis, then you may have too many (perhaps dependent) Cost or Risk figures of merit and not enough Performance figures of merit. Usually the Do Nothing alternative has low cost and little risk. This hints that perhaps we should have a similar number of Performance, Cost and Risk figures of merit.

If a nihilistic Do Nothing alternative attains one of the highest scores in the tradeoff analysis, then you might have too few Performance figures of merit. For example, in a tradeoff study for a municipal transportation system, if a pogo stick wins, then you have missed some Performance figures of merit. As a second example, suppose a young couple on a date is selecting a movie to go to and they define their alternatives as comedy, romance, blood and guts, and do not go to the movies. If the Do Nothing alternative wins, then they missed some Performance figures of merit.

Just as you should not add apples and oranges, you should not combine Performance, Cost, Schedule, and Risk FoMs. All the Performance FoMs should be combined; then all the Cost FoMs should be combined; then all the Schedule FoMs should be combined; and finally all the Risk FoMs should be combined. Then the Performance, Cost, Schedule, and Risk combinations can be combined with clearly stated weights, ¼, ¼, ¼, ¼, is a good default. If the Do Nothing alternative still wins, then you may have the weight for Performance too low.

7. SENSITIVITY ANALYSIS

In many instances, a system design specification can be inspected to reveal that the major attributes of the system (such as performance, cost, schedule, or risk) are driven by relatively few parameters. A rough interpretation of Pareto's rule would state that 80% of the influence can be traced to 20% of the parameters. That is, variations of the few prominent parameters can have a substantial impact on the characteristics of the system. Information of this type can be extremely valuable to the stakeholder because it not only alludes to factors that are central in characterizing the system, but it also conveys to the stakeholder which parameters are robust to change. This can shed new light on tradeoff possibilities. The process of uncovering parameters that drive the system's properties is called *sensitivity analysis*.

We suggest that all tradeoff studies should include a sensitivity analysis. The results of a sensitivity analysis can be used to (1) validate a model, (2) warn of strange or unrealistic model behavior, (3) suggest new experiments or guide future data collection efforts, (4) point out important assumptions of the model, (5) suggest the accuracy to which the parameters must be calculated, (6) guide the formulation of the structure of the model, (7) adjust numerical values for the parameters, (8) allocate resources, and (9) point out the true cost drivers [Karnavas, Sanchez, and Bahill, 1993]. If you show your customer the requirements that are driving the cost of the system, then he or she may relax a requirement and save a lot of money.

There are many methods for carrying out sensitivity analyses such as relative-sensitivity measures, Response Surface Methodology, Sinusoidal Variation of Parameters, etc. In analyzing the sensitivities of tradeoff study parameters, typically we are interested in (1) those parameters and Figures of Merit that are most important and deserve further attention and (2) those parameters that, when varied, could change the recommended alternative. The first issue could be investigated using a relative-sensitivity measure for each parameter for each alternative. The second could be ascertained by employing a search algorithm using the sensitivity of each parameter [Karnavas, Sanchez, and Bahill, 1993]. Typical tradeoff study parameters include weights at all levels of the tradeoff structure, and the scoring function inputs and parameters.

Kirkwood [1997] gives a nice discussion of how sensitivity analyses can be performed using a simple spreadsheet. For doing sensitivity analyses on the weights in a tradeoff study, Kirkwood [1997] gives a useful weight-ratio formula for determining what the values should be for the fixed weight parameters as the variable weight is varied.

8. DISCUSSION

So, what does this all mean? Suppose, through conducting a tradeoff study as prescribed, we obtain the following results:

(a) Alternative A score = 0.9,
(b) Alternative B score = 0.89,
(c) Alternative C score = 0.5,
(d) Alternative D score = 0.4.

Does this mean that alternative A is preferred to alternative B? No. It means that alternatives A and B are the preferred choices. It is up to the stakeholder to distinguish between close results. We are not able to make precise judgments because:

1. Our method for deriving weights is ad hoc and not based upon axioms.
2. We only mention a few combining methods and we don't know what type of problem is best suited for each method.
3. We have ignored interactive effects among FoMs. For a discussion of interactions among FoMs, see Daniels, Werner, and Bahill [2000].

The value of a tradeoff study is not in its mathematical elegance or precision. Its value is in stating the problem in a public and verifiable manner.

9. SUMMARY

Tradeoff studies are important because:

1. They create an objective mechanism for evaluating systems.
2. They document the decision process.
3. The preference structure is quantified.
4. The tradeoff study process educates the customer.
5. They help validate system requirements by providing a measurable quantity that helps determine if and how well a design satisfies the requirements [Grady, 2000].
6. They assist in selecting the preferred alternative.

The purpose of this paper is to provide the practicing engineer with an introduction to tradeoff analysis theory and to demonstrate the benefits these analyses can offer. Tradeoff analyses give systems engineering the power to confidently make design decisions by structuring the design process through an analytical mapping with stakeholder input. Since tradeoff analyses are performed early on in the design process, the expensive consequence of misinterpreting or not validating vital customer requirements in the deployment or implementation phase can be avoided. By such rigorous upfront analysis of the stakeholders' requirements and preferences, we can be sure that the system adopted to satisfy the stakeholders' objectives will effectively do so throughout its lifecycle.

We have exploited the hierarchical nature of systems design by approaching the discussion of tradeoff analyses and ultimately scoring functions through selective decomposition of the system design process. The SIMILAR process suggests that effective system design is achieved in a hierarchical and iterative fashion. A critical subfunction of the system design process is the investigation of alternative designs. Investigating alternative designs is achieved by first deriving an appropriate Figure of Merit architecture based on customer requirements and preferences. Individual FoMs are evaluated by running models or acquiring measurements on the actual system. The evaluation data are then transformed into normalized numerical scores using respective standard scoring functions that capture the customers' relative preferences. Scoring functions allow comparisons to be carried out across the many diverse FoMs used for evaluating a given alternative. Comparisons of alternative system designs involve combining scoring function outputs in such a way that decisions regarding preferred system alternatives can be made and defended while concurrently preserving customer preferences and requirements.

The most important result of a tradeoff study is not determining the preferred alternative. The most important product of a tradeoff study is the documentation, with the FoMs, the scoring functions, their parameters, and the input values that were used. This lets everyone see and understand why the preferred alternative was preferred.

ACKNOWLEDGMENTS

We thank Dennis Buede and Wayne Wymore for comments on the manuscript. The authors gratefully acknowledge the support of the Federal Aviation Administration (AAR-424). This paper is based on work done at The University of Arizona and Sandia

National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

A.T. Bahill and F.F. Dean, "Discovering system requirements," Handbook of systems engineering and management, A.P. Sage and W.B. Rouse (Editors), Wiley, New York, 1999, pp. 175–220.

A.T. Bahill and B. Gissing, Re-evaluating systems engineering concepts using systems thinking, IEEE Trans Syst Man Cybernet SMC-28 (1998), 516–527.

A.T. Bahill, S.O. Dahlberg, and R.A. Lowe, Difficulties in using multicriterion decision making techniques for tradeoff studies selecting amongst alternative concepts, Proc 8th Annu Sym Int Council Syst Eng (INCOSE), August 3–7, 1998, Vancouver, Canada, pp. 165–170.

A. Bardossy and L. Duckstein, Fuzzy rule-based modeling with applications to biological and engineering systems, CRC Press, Boca Raton, FL, 1995.

B.S. Blanchard and W.J. Fabrycky, Systems engineering and analysis, Prentice-Hall, Upper Saddle River, NJ, 1998.

K. Borcherding, T. Eppel, and D. von Winterfeldt, Comparison of weighting judgments in multiattribute utility measurement, Manage Sci 37 (1991), 1603–1619.

B.G. Buchanan and E.H. Shortliffe, Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading MA, 1984.

D. Buede, The engineering design of systems, Wiley, New York, 2000.

W.L. Chapman, A.T. Bahill, and W.A. Wymore, Engineering modeling and design, CRC Press, Boca Raton, FL, 1992.

J.A. Cooper, Soft mathematical aggregation in safety assessment and decision analysis, Proc 17th Int Syst Safety Conf, August 16–21, 1999, Orlando, FL.

J.D. Daniels, P.W. Werner, and A.T. Bahill, Identifying and modeling safety hazards, Proc 10th Annu INCOSE Int Symp, July 16–20, 2000, Minneapolis, MN, pp. 107–114.

L. Duckstein, W. Treichel, and S. El Magnouni, Multicriteria analysis of groundwater management alternatives, Cahier No 114, Cahier du Lamsade, Université de Paris-Dauphine, Place du Marechal de Lattre de Tassigny, Paris, 1993.

W. Edwards, How to use multiattribute utility analysis for social decision making, IEEE Trans Syst Man Cybernet SMC-7 (1977), 326–340.

W. Edwards (Editor), Utility theories: Measurements and application, Kluwer Academic, Boston, 1992.

A. Goicoechea, D.R. Hansen, and L. Duckstein, Multiobjective decision analysis with engineering and business applications, Wiley, New York, 1982.

J.O. Grady, System engineering deployment, CRC Press, Boca Raton, FL, 2000.

R.A. Howard, "In praise of the old time religion," Utility theories: Measurements and applications, W. Edwards (Editor), Kluwer Academic, Boston, 1992, pp. 27–55.

W. J. Karnavas, P. J. Sanchez, and A. T. Bahill, Sensitivity analyses of continuous and discrete systems in the time and frequency domains, IEEE Trans Syst Man Cybernet SMC-23 (1993), 488–501.

D.L. Keefer, and S.M. Pollock, Approximations and sensitivity in multiobjective resource allocation, Oper Res 28 (1980), 114–128.

R.L. Keeney and H. Raiffa, Decisions with multiple objectives: Preferences and value tradeoffs, Wiley, New York, 1976.

R.L. Keeney, Value-focused thinking: A path to creative decision-making, Harvard University Press, Cambridge, MA, 1992.

C.W. Kirkwood, Strategic decision making: Multiobjective decision analysis with spreadsheets, Duxbury Press, Belmont, CA, 1997.

C.W. Kirkwood and R.K. Sarin, Preference conditions for multiattribute value functions, Oper Res 28 (1980), 225–232.

J.A. Moody, W.L. Chapman, F.D. Van Voorhees, and A.T. Bahill, Metrics and case studies for evaluating engineering designs, Prentice Hall PTR, Upper Saddle River, NJ, 1997.

T.L. Saaty, The analytical hierarchy process, McGraw-Hill, New York, 1980.

R.E. Steuer, Multiple criteria optimization: Theory, computation, and application, Wiley, New York, 1986.

F. Szidarovszky, M. Gershon, and L. Duckstein, Techniques for multiobjective decision making in systems management, Elsevier, Amsterdam, 1986.

J. Von Neumann and O. Morgenstern, Theory of games and economic behavior, Princeton University Press, Princeton, NJ, 1947.

S.R. Watson and D.M. Buede, Decision synthesis: The principles and practice of decision analysis, Cambridge University Press, Cambridge, 1987.

W.L. Winston, Operations research: Applications and algorithms, Duxbury, Belmont, CA, 1994.

A.W. Wymore, Model-based systems engineering, CRC Press, Boca Raton, FL, 1993.

Jesse Daniels is a systems engineer with BAE Systems in San Diego. He did this work as a graduate student at the University of Arizona in the Systems and Industrial Engineering Department where he received his B.S. and M.S. in systems engineering in 1999 and 2000.

Paul W. Werner is a Principal Member of the technical staff at Sandia National Laboratories. He works with the High Consequence Surety Engineering Department. He received his Ph.D. in electrical engineering from the University of New Mexico in 1991. He is also a Commander and a Naval Flight Officer in the U.S. Naval Reserve.

A. Terry Bahill is a Professor of Systems Engineering at the University of Arizona and an Engineering Fellow with Raytheon Missile Systems in Tucson. He received his Ph.D. in electrical engineering and computer science from the University of California, Berkeley, in 1975. He holds a U.S. Patent for the Bat Chooser™, a system that computes the Ideal Bat Weight™ for individual baseball and softball batters. He is Editor of the CRC Press Series on Systems Engineering. He is a Fellow of the Institute of Electrical and Electronic Engineers (IEEE) and of the INCOSE. He is the chair of the INCOSE Fellows Selection Committee.

Corrigendum

Correction to "Quantitative Methods for Tradeoff Analyses"

A. Terry Bahill and Jesse Daniels

[1]BAE SYSTEMS, *San Diego* CA 92127 (*terry.bahill@BAEsystems.com jesse.daniels@BAEsystems.com*)

Received 21 September 2004
Published online in Wiley InterScience (www.interscience.wiley.com).
DOI 10.1002/sys.20019

There are many popular methods for combining data in a tradeoff study. One of these is forming the product of the scores and weights [Daniels, Werner and Bahill, 2001]. Unfortunately, in this publication the printed formula was wrong.

The equation should have the weights in the exponent, like this:

$$f = \prod_{i=1}^{n} x_i^{w_i}$$

In this equation, n represents the number of Figures of Merit (FoMs) that are to be combined, x_i represents the output of the scoring function for the i^{th} FoM and w_i represents the weight of importance assigned to the i^{th} FoM. The Product Tradeoff Function is commonly used for example in computing cost to benefit ratios and in doing risk analyses.

Unfortunately, the published equation on page 195 multiplied the weight times the score, like this:

$$f = \prod_{i=1}^{n} x_i w_i$$

Formulated like this, the weights have absolutely no effect. We apologize for any confusion this may have caused.

Acknowledgment

We thank Harley Henning for pointing out this typographical error.

REFERENCE

J. Daniels, P. W. Werner and A. T. Bahill, "Quantitative Methods for Tradeoff Analyses," System Engineering, 4(3): 190–212, 2001.