1.3 Drawbacks of Simple Additive and Multiplicative Scoring and Utility Models

Everything should be made as simple as possible, but not simpler.
—A. Einstein

Simplicity is a good thing only for those who know and never cross the border line between simplicity and oversimplification. Unfortunately, it is in human nature to trade accuracy for simplicity. Simplicity at any cost is incompatible with professional problem solving. Simple scoring techniques are the area of decision making where accuracy is traded for simplicity. More often than not, the simple scoring models yield oversimplifications and questionable results. This is the reason why simple scoring and utility models are generally *not acceptable* in professional evaluation. In this chapter, our goal is to explain and prove this claim.

The simple scoring methods are used to score individual components of complex systems, and then to evaluate and compare competitive systems using a weighted mean of component scores [KLE78, KLE80]. The most important such approach is the *simple additive scoring* (SAS, also called simple additive weighting, SAW [TZE11], or weighted linear combination, WLC [MAL99]) that has been used for many years. SAS models for evaluation (e.g., [SCH69, SCH70, MIL66, MIL70, YOO95, TRI00, PAL02]) assume that the quality of an object is a weighted sum of the quality indicators of its components/attributes, as follows:

$$\text{Overall\_score} = \sum(\text{component\_weight} \times \text{component\_score})$$

In a similar way, *simple multiplicative scoring* (SMS) (e.g., [WHI63, CHE92, CHA01, YOO95, TRI00, VAN04]) uses a geometric mean model with a reasonable intention to more heavily penalize objects with poor component values:

$$\text{Overall\_score} = \prod(\text{component\_score})^{\text{component\_weight}}$$

Each component weight denotes the relative importance of the evaluated component, and the component score reflects the quality of evaluated

Soft Computing Evaluation Logic: The LSP Decision Method and Its Applications,
First Edition. Jozo Dujmović.

Companion website: www.wiley.com/go/Dujmovic/Soft_Computing_Evaluation_Logic

component. The overall score is then used for decision making (evaluation, comparison, and selection of the best alternative).

It is self-evident that the next step toward stronger penalizing objects with poor components could be based on the harmonic mean:

$$\text{Overall\_score} = 1/\left[\sum(\text{component\_weight}/\text{component\_score})\right]$$

However, while the simple additive scoring is by far the most popular, and the multiplicative scoring is much less frequently discussed, it is interesting that the harmonic mean model seems to be completely neglected by the proponents of fixed (additive or multiplicative) scoring. Therefore, we will now focus on properties of additive and multiplicative scoring.

1.3.1 Simple Additive Scoring: The Irresistible Attractiveness of Simplicity

The SAS model can be mathematically written as follows:

$$S_0 = A(S_1,\ldots,S_n) = \sum_{i=1}^{n} W_i S_i,\ W_1 + \cdots + W_n = 1,\quad W_i > 0,\quad i = 1,\ldots,n.$$

Here n denotes the number of inputs, $W_1, \ldots, W_n$ are normalized weights, $S_1, \ldots, S_n$ are attribute scores, and S_0 is the resulting overall (output) score. The scores can be in any range, and we will assume the unit interval $S_i \in I = [0,1],\ \ i = 0,\ldots,n.$ Since $W_1 + \cdots + W_n = 1,$ the average weight is $\overline{W} = (W_1 + \cdots + W_n)/n = 1/n.$ In other words, the fundamental property of additive scoring models is that the average relative importance of a component must decrease when the number of components increases.

Another view of this property can be obtained by computing the average impact of SAS inputs. The maximum impact of input S_i is the difference between output scores obtained for the extreme cases $S_i = 1$ and $S_i = 0$:

$$\delta_i = A(S_1,\ldots,S_{i-1},1,S_{i+1},\ldots,S_n) - A(S_1,\ldots,S_{i-1},0,S_{i+1},\ldots,S_n) = W_i.$$

Consequently, following is the average impact of an input of SAS:

$$\bar{\delta} = (\delta_1 + \cdots + \delta_n)/n = (W_1 + \cdots + W_n)/n = 1/n.$$

As n increases, the average impact decreases, and for larger values of n the impact becomes negligible. In other words, for large/complex systems[1] the

1 Complex systems with large number of attributes are easily found in the area of software. For example, evaluation models of operating systems and database systems typically include several hundred (typically 300–600) elementary attributes. Even a detailed car characterization model can have more than 200 attributes.

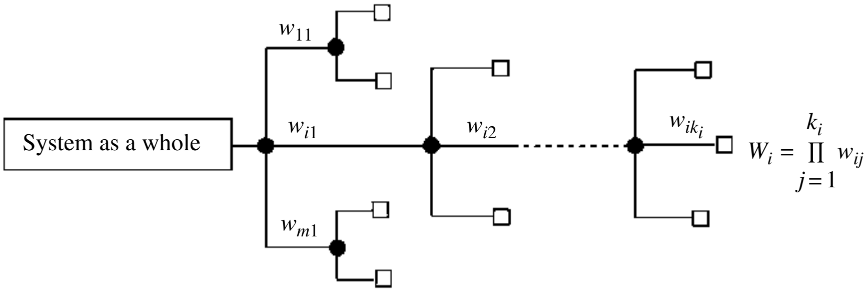

Figure 1.3.1 Hierarchical decomposition of weights.

number of attributes is also large, and each individual attribute automatically becomes insignificant. That is not a desirable property because of two main reasons: (1) it limits the number of inputs, and (2) there are critical attributes that must have significant impact regardless the number of inputs.[2] In addition, if a model is inappropriate for large n that does not implies that it is appropriate for small n.

The weights $W_1, \ldots, W_n$ are computed by decomposing the attribute tree as illustrated in Fig. 1.3.1. At the initial (root) level, the system as a whole is decomposed into m major components having individual weights (relative importance) $w_{11}, \ldots, w_{m1}$, $w_{11} + \cdots + w_{m1} = 1\left(\forall i, j, \quad w_{ij} > 0\right)$. This process systematically repeats for each component that can be further decomposed. The result is a tree structure where each branch has a weight and leaves denote elementary input attributes (performance variables). The resulting weights of attributes are obtained as products of weights along the path of k_i branches that connect a selected i^{th} leaf with the root of the weight decomposition tree:

$$W_i = \prod_{j=1}^{k_i} w_{ij}, \quad \sum_{i=1}^{n} W_i = \sum_{i=1}^{n} \prod_{j=1}^{k_i} w_{ij} = 1.$$

These weights are then used in the SAS formula, and if elementary scores are normalized so that $0 \le S_i \le 1, \quad i = 1, \ldots, n$, then the overall score is also normalized: $0 \le S_0 \le 1$. The zero score denotes worthless (or nonexistent) component, and the score 1 (or 100%) denotes a perfect satisfaction of the evaluator's requirements. Additive models are particularly popular and carefully studied in utility theory [FIS64, FIS70], and its applications [EDW82].

2 In a simple example of a family car evaluation, if a family has six members, all cars with less than six seats should have zero score for the number of seats and should be rejected, regardless how many attributes are used for car evaluation and how good the other attributes are.

The main advantage of SAS is its simplicity, and this is the reason for its frequent use in a variety of evaluation problems. However, additive scoring has many disadvantages [DUJ72a, DUJ72b] that make this method unsuitable in a general case of professional evaluation of complex systems. Following is a summary of eight main SAS drawbacks.

(I) Limited Number of Components for Evaluation

If the evaluation model includes n input attributes, then the average weight (as well as the average impact) of each component is $1/n$. For example, in the case of 100 attributes, the average importance of each attribute is only 1%. In such a case, even the most important components change the final score for only a few percent, and many components affect the final score for only a negligible fraction of percent. This yields insufficient sensitivity to many relevant features and practically limits the number of components that can be evaluated.

A very convincing example of this drawback can be found in [GIL76] (MECCA method, same as Weighted Ranking by Levels [SCH69, SCH70]). In his model of IBM VM/CMS-TSO evaluation (actual MECCA evaluation of IBM VM/CMS and TSO software from a Danish user), Gilb suggests the following five-level decomposition path from the item called "Test Aids" to the leaf "Task Scheduling" (weights are expressed as percentages):

1. Test Aids [50%]
    11. Applications Development [50%]
        115. OS Compatibility [15%]
            1152. Limitations of it [30%]
                11523. Task Scheduling [10%]

Therefore, the weight of the leaf #11523 (Task Scheduling) is $0.5 \times 0.5 \times 0.15 \times 0.3 \times 0.1 = 0.001125$ (i.e., 0.1125%). This result should normally trigger several self-evident questions:

1) Why should we consider an insignificant component that causes an overall effect that is less than or equal to only 0.1125%?
2) How can we justify the cost of evaluating components that are insignificant and therefore easily negligible?
3) Mathematical models should have a concordant (balanced) level of accuracy of inputs and parameters. In [GIL76], the weights predominantly change in steps of 5% to 10%, suggesting that this is the highest level of accuracy that the evaluator can provide. From that standpoint, it is not reasonable to make efforts that support precision at the level of 0.1125%. How can this obvious discrepancy be justified?

Unfortunately, neither the author nor the reviewers of [SCH69, SCH70, GIL76] asked and answered these questions and similar obvious questions presented below.

(II) Evaluation Effort Wasted

Professional evaluation of complex systems can be a significant and expensive effort. The total evaluation effort (and cost) depends primarily on the number of attributes and the complexity of attribute evaluation. Evaluation of attributes that have negligible effect on the final result can be considered a waste of resources.

The attribute evaluation effort is distributed in a range from low-effort to high-effort attributes. A typical example of a low-effort attribute (in the case of car evaluation) is the power of engine. It is a well-known quantitative attribute, and its scoring can be based on a simple scoring function (Fig. 1.1.2). For more complex attributes, such as attributes related to a complex computer operating system (OS), the attribute evaluation may need significant effort, for both the development of an attribute evaluation criterion and the attribute evaluation process. Qualitative indicators, such as the above example of "OS compatibility," require a careful expert study of the OS implementation and properties. Such indicators cannot be reliably scored without a detailed and time-consuming expert investigation of all aspects of the evaluated attribute.

Similarly, there are quantitative attributes that are in the high-effort category. For example, early computer benchmarking studies [SHA69, BEN75, CUR84, FER78a, FER78b, NBS77] report examples of very high cost of mainframe computer performance measurements. At the present time, the cost of benchmarking is significantly reduced; nevertheless, any nonstandard evaluation of a response time of a computer system usually requires a time-consuming and expensive development of a test workload, as well as a subtle measurement procedure. Similarly, in the evaluation of ecological systems, measurements of attributes frequently require time consuming field trips of expert personnel. Therefore, it is only reasonable to invest the evaluation effort and resources in evaluation of attributes that sufficiently contribute to the overall evaluation result. SAS models with low-impact attributes cannot give good reason for expensive evaluation procedures. Unfortunately, some attributes become insignificant by the additive nature of the SAS model, and not because of evaluator's decision about negligible importance. If low-impact attributes are not eliminated then the effort needed for evaluation of low-impact attributes becomes unjustified and wasted. Generally, it is not reasonable to feed expensive and highly precise inputs into a very imprecise decision model based on simplistic additive scoring. The sophistication of decision models must match the sophistication of selecting attributes and their measurement and evaluation.

(III) Impossibility to Model Mandatory Requirements

SAS models do not support mandatory requirements since the absence of a necessary feature cannot decrease the overall score for more than the relative weight of that feature. Indeed, if a system has the score $S_0 = W_1S_1 + \cdots + W_iS_i + \cdots + W_nS_n$, then the difference between the cases where $S_i = 1$ and $S_i = 0$ is only W_i (and this is typically a small value, on the average *1/n*). Unfortunately, we frequently have situations where the total absence of a vital feature cannot be compensated by other inputs (i.e., it must generate the zero overall score). For example, a computer system with insufficient memory or insufficient processor speed cannot satisfy the requirements of a demanding user regardless the quality of other components. In such cases, the mathematical model must provide a possibility to properly penalize (to the level of rejecting) all competitors with insufficient satisfaction of vital requirements. Mathematically speaking, the corresponding aggregation function must support the annihilator 0.[3] Additive scoring models lack this very important feature that is easily observable and frequently present in intuitive evaluation. In the majority of serious applications, this drawback is sufficient to eliminate SAS as an acceptable mathematical model. By definition, *all SAS inputs are always optional.*

(IV) Impossibility to Model the Simultaneity of Requirements

SAS models cannot express graded simultaneity (the need for simultaneous satisfaction of several requirements using adjustable degree of simultaneity). The additivity assumes independence of attributes where a limited decrement $\Delta S_i \leq \Delta S_{ij}$ of score S_i can always be compensated by an appropriate increment (ΔS_j) of *any* other score S_j, as long as $W_i(S_i - \Delta S_i) + W_j(S_j + \Delta S_j) = W_iS_i + W_jS_j$ or $W_i\Delta S_i = W_j\Delta S_j$. Since $S_j + \Delta S_j \leq 1$ and $W_i\Delta S_{ij} = W_j\Delta S_j$, it follows that the maximum decrement of score S_i that can be compensated by score S_j is $\Delta S_{ij} = W_j(1 - S_j)/W_i$. Consequently, the additive scoring is a model that has significant (nonadjustable) compensative properties. SAS does not sufficiently support penalizing systems that lack simultaneity in satisfaction of crucial components. So, the additive scoring is inappropriate in all cases where the mathematical model must be able to provide an adjustable level of penalizing systems with insufficient simultaneity in satisfaction of important requirements. For example, in the case of computer evaluation, we *simultaneously* need both quality hardware and quality software. However, if we use the SAS model, the lack of a software component can be compensated by an unrelated hardware property, such as large disk capacity, and vice versa. Additive scoring does not provide an appropriate way to require simultaneous presence of necessary software and hardware features because the constant overall andness of arithmetic mean is only 50%.

---

3 A logic function $F(x_1, \ldots, x_n)$ has an annihilator a if the condition $x_i = a,\ i \in \{1,\ldots,n\}$ causes $F(x_1,\ldots,x_n) = a$. The most important annihilators are $a = 0$ and $a = 1$.

(V) Impossibility to Model Sufficient Conditions and Substitutability

Simultaneity and substitutability are symmetric (dual) properties and all SAS problems with mandatory requirements and simultaneity are mirrored as problems with sufficient requirements and substitutability. Indeed, SAS models cannot properly express substitutability: the absence of a feature always decreases the overall score and cannot be *fully compensated* by the availability of some other equivalent feature. For example, if i and j are indices representing two fully substitutable alternative ways to satisfy a specific requirement, then only one of them should be sufficient for the complete satisfaction of user's needs. However, the additive scoring expression $W_iS_i + W_jS_j$ assumes that both S_i and S_j are needed to achieve the maximum overall score. The level of replaceability must be adjustable, and this is not supported by the additive scoring models. The relationship $(x+y)/2 = [(x \wedge y) + (x \vee y)]/2$ indicates that the arithmetic mean is located halfway between conjunction and disjunction, i.e., it is equally unsuitable to model simultaneity and substitutability. Both the global andness and the global orness of the arithmetic mean are only 50%. Consequently, both the sufficient conditions and the graded substitutability cannot be modeled using SAS models.

(VI) Impossibility to Model Asymmetric Logic Relationships

Human evaluation reasoning very frequently includes asymmetric criteria where some attributes are mandatory and some are optional. For example, in the evaluation of suitability of the house location a homebuyer may decide that the proximity to work and the proximity to school for children are *mandatory requirements* and the proximity to post office and public swimming pool are desired but not mandatory (i.e., these are *optional requirements*). In this example, the unacceptable proximity to work or school should yield the zero overall score, while the unacceptable proximity to post office or swimming pool should yield a controlled decrease of the overall score, but not the zero overall score and the rejection of the evaluated house. Generally, mandatory attributes must support the annihilator 0 and optional attributes must not support the annihilator 0; this property cannot be achieved using SAS. SAS models use weights as the *only means* for differentiating the role of inputs. This is not sufficient for expressing asymmetric logic relationships, such as a combination of mandatory, desired, and optional features, or a combination of sufficient and optional conditions. SAS attributes can be more or less important, but they cannot be necessary or sufficient.

(VII) Inappropriate Optimum Systems

In many cases, a component score S_i is a limited linear function of component cost C_i:

$$S_i(C_i) = \min(1,\ a_iC_i) = \begin{cases} a_iC_i, & 0 \le C_i \le 1/a_i \\ 1, & C_i \ge 1/a_i \end{cases},\quad a_i = const.,\ 0 \le S_i \le 1.$$

A typical example of such a function is the capacity of computer memory: memory blocks have a constant price, and by adding more blocks, we can linearly increase both the total cost and the user satisfaction. Users regularly have limited needs, and when the need for memory is completely satisfied, further addition of memory has no effect. The corresponding overall score and the overall cost are:

$$S_0 = \sum_{i=1}^{n} W_i S_i(C_i) = \sum_{i=1}^{n} A_i C_i, \quad A_i = a_i W_i,$$

$$C_0 = \sum_{i=1}^{n} C_i, \qquad 0 \le C_i \le 1/a_i.$$

Let us now determine the optimum configuration in the case where the available resources are limited to C. So, we have to distribute the total amount C in a way that maximizes the overall score:

$$S_0^{\max} = \max_{C_1 + \ldots + C_n \le C} \sum_{i=1}^{n} A_i C_i.$$

The additive components can come in any order, and let us assume that the cost weights are sorted so that $A_1 \ge A_2 \ge \ldots \ge A_n$. The *optimum configuration* is one that achieves the maximum overall score for a constrained overall cost. It is based on the following resource allocation procedure: Buy the maximum quantity of components according to the decreasing cost weight order, and stop when the available resources are exhausted. In other words, the resulting *optimum strategy* is:

$$C_i = C_i^{\max} = 1/a_i, \quad i = 1,2,\ldots,k-1$$

$$C_k = C - \sum_{i=1}^{k-1} C_i, \quad 0 < C_k \le 1/a_k = C_k^{\max}$$

$$C_i = 0, \quad i = k+1,\ldots,n$$

$$S_0^{\max} = \sum_{i=1}^{k-1} A_i C_i^{\max} + A_k C_k = \sum_{i=1}^{k-1} W_i + A_k C_k.$$

For example, let $S_0 = 0.4S_1 + 0.3S_2 + 0.2S_3 + 0.1S_4$, and $S_1 = C_1$, $(0 \le C_1 \le 1)$, $S_2 = 2C_2$, $(0 \le C_2 \le 0.5)$, $S_3 = 4C_3$, $(0 \le C_3 \le 0.25)$, $S_4 = 5C_4$, $(0 \le C_4 \le 0.2)$. Consequently, $S_0 = 0.8C_3 + 0.6C_2 + 0.5C_4 + 0.4C_1$ and $0 \le C \le 1.95$. If the available resources are $C = 0.9$, then the optimum solution is $C_3 = 0.25, C_2 = 0.5$, $C_4 = 0.15$, $C_1 = 0$, $S_0^{\max} = 0.575$. So, the first component (which is both the most important and the most expensive) will be missing (because its inclusion would decrease the value of $S_0^{\max}$), regardless of the clear indication that this component might be vital for the proper functioning of the analyzed system.

The resulting policy is an extreme "first things first" approach: Allocate funding to achieve the maximum satisfaction of the highest cost weight requirement, then the maximum satisfaction of all requirements in the decreasing cost weight order. When the resources are exhausted, select zero satisfaction of all remaining requirements regardless of their importance. Obviously, such "optimum systems" can easily miss some vital system components (e.g., an optimum computer configuration might have the fastest processor, the largest monitor, and the maximum disk capacity, but no memory and no data communication units). Because of the linear character of the SAS model, the resulting optimum configuration is obviously wrong, and is sufficient to disqualify the SAS method.

(VIII) Preferential Independence is Rarely Present

The underlying assumption of SAS models is the preferential independence [KEE76, OLS96, YOO95]. There are various definitions of preferential independence [KEE76, BEL02]. The simplest informal definition is that the contribution of an individual input to the overall score is independent of other input values [YOO95]. In other words, in the case of preferential independence, $\partial S_0/\partial S_i$ is not a function of S_j, $j \neq i$, and for additive models $\partial S_0/\partial S_i = const$. Three simple examples where the preferential independence does not hold are shown in Table 1.3.1.

In the first case, the patient disability is a function of motor symptoms (e.g., inability to walk, or difficulties in using hands) and of sensory symptoms (e.g., pain). Obviously, a patient is disabled if either motor or sensory problems are present (disjunctive criterion). The effect of motor symptoms on disability is different when the pain is present and when the pain is absent. When the motor symptoms are present, the disability is already there and the pain contributes to the overall degree of disability less than in cases where the pain affects the disability alone.

In the second case, we have a conjunctive criterion where the evaluator wants simultaneously the powerful engine, good comfort, safety, and appearance. If the comfort, safety, and appearance are low, then the overall score will be low and the effect of the power of engine might be negligible. If the comfort, safety, and appearance are high, then the engine power is a bottleneck that

Table 1.3.1 Sample cases where preferential independence does not hold.

Evaluated system	Input 1	Input 2	Input 3	Input 4
Patient disability	Motor symptoms	Sensory symptoms		
Car	Power of engine	Comfort	Safety	Appearance
Home	Home location	Home quality		

has a decisive impact on the overall score. Similarly, if a home location is unacceptable, the home quality has a lower effect on the homebuyer's satisfaction than in cases of good locations. Therefore, in both conjunctive and disjunctive cases, there is no preferential independence that would warrant the use of additive scoring.

Another aspect of preferential independence is defined as the tradeoff (the substitution rate) of S_i and S_j, $j \neq i$ that is independent of all other scores S_k, $k \neq i$, $k \neq j$ [KEE76]. In the above car evaluation example, the high engine power of a sports car could partially compensate deficiencies in comfort, provided that safety and appearance are good. That might be impossible in cases of poor safety and appearance that already disqualify the car.

Human evaluation reasoning frequently includes conjunctive and disjunctive criteria that request simultaneity or substitutability of inputs. In all such cases, there is no preferential independence and the additive scoring in not appropriate. Consequently, SAS is a verification that everything should be as simple as possible, but not simpler. Simplicity (as the only reason and sufficient justification for SAS as a preferred method for solving delicate evaluation problems in social sciences) is clearly and explicitly visible in [EDW82, p. 74]: "Although the literature describes very complicated aggregation rules, we use only one because it is by far the simplest. The equation takes the following form: $U_j = \sum_{i=1}^{n} w_i u_{ij}$ ...." This method (called *multiattribute utility technology*) is introduced in [EDW82] by the series co-editor as "an appropriate way to go about a rigorous, quantitative assessment of social programs." Without apology, let us say that these statements are similar and equally unacceptable as the following one: "In mathematics, there are many different curves, but we use only the straight line because it is by far the simplest."

Oversimplified SAS models are frequent in GIS literature and justified by the fact that they are "easy-to-understand and intuitively appealing to decision-makers" [MAL06]. For example, urban development is not possible on excessive slopes of terrain. However, SAS models show positive suitability for building homes even on vertical slopes provided such locations satisfy some other requirements (e.g., the proximity to roads), what is meaningless. Chapter 4.6 and [DRA17] show examples of easily visible errors in SAS-generated suitability maps.

Our analysis convincingly disqualifies SAS in all applications where evaluation models must be compatible with human reasoning and decision making. Human mind is not a simple linear machine, and the unsuitability of SAS models is easily visible and provable. Consequently, it is necessary to explain where the high popularity of SAS comes from.

The grossly exaggerated popularity of SAS can be explained in three ways. First, it is in human nature to try to simplify problems, and oversimplification is just one little additional step in this attractive direction. Second, the majority of high-level corporate, public, military, social, and political decisions are done by people for whom, by the generalist nature of their job, everything beyond addition is simply "too mathematical." Third, there are cases where SAS is an appropriate evaluation technique (as in the case of using GPA for ranking of students), and such cases are then unjustifiably generalized.

1.3.2 Simple Multiplicative Scoring

The SMS model is a weighted geometric mean:

$$S_0 = \prod_{i=1}^{n} S_i^{W_i}, \ W_1 + \cdots + W_n = 1, \quad W_i > 0, \ i = 1, \ldots, n.$$

The multiplicative nature of this scoring model causes high importance of low input scores. Even one low input score can significantly reduce the value of the overall score S_0.

The following drawbacks of the multiplicative model are similar to the drawbacks of the additive model, and we are going to list them without detailed exemplifications.

(I) Impossibility to Model Requirements for Nonmandatory (Optional) Inputs

SAS models do not support mandatory inputs, and SMS models do not support optional inputs. If a system has the overall score $S_0 = S_1^{W_1} \cdots S_n^{W_n}$ then we have $S_0 = 0$ in all cases where $S_i = 0, \ i \in \{1, \ldots, n\}$. In other words, *all inputs are always mandatory*. Since there are no nonmandatory/optional inputs, if the most insignificant attribute of an evaluated system is not satisfied, such a system must be rejected. Obviously, this is an extreme requirement that is generally not acceptable.

(II) The Contradiction of Insignificant but Mandatory Inputs

Similar to additive models, if the number of inputs n increases then the weights must decrease, making some inputs completely insignificant. Unfortunately, due to the multiplicative nature of the model all inputs are mandatory. That creates inconsistent and contradictory conditions: The fact that each input is mandatory implies the significance of all inputs, while unavoidable low weights of some inputs imply their insignificance. Generally, there is no way to properly balance these inconsistent and contradictory standpoints. In Chapter 2.2, we show that the consequence of this contradiction is hypersensitivity, i.e., mathematical models where negligible increments of insignificant input attributes produce huge increments of the overall score (a property that is not observable in human evaluation reasoning).

(III) Incompleteness of Evaluation Models

The only way to use SMS models is to omit *all* nonmandatory requirements. Consequently, SMS models are incomplete because they include only the "tip of the iceberg," i.e., essential mandatory requirements—and omit all other contributing components that are classified as optional but not insignificant.

(IV) Impossibility to Adjust the Degree of Simultaneity of Requirements

SMS models significantly penalize systems that have any poor components. However, SMS models use a fixed level of hard simultaneity (as shown in *Part Two*, the global andness of geometric mean is relatively low, approximately 67%). SMS models cannot express frequently needed levels of simultaneity that are either greater than, or less than the simultaneity implemented by the geometric mean. In particular, the soft simultaneity that does not support annihilator 0 cannot be modeled using SMS.

(V) Impossibility to Model Adjustable Substitutability and Sufficient Conditions

Neither adjustable substitutability nor sufficient conditions can be expressed by the SMS models. The geometric mean is conjunctively polarized and cannot model any significant level of disjunctive polarization. It is important to note that this drawback is more exposed and critical for SMS then for SAS, because the SMS andness of 67% is further from the andness range for substitutability $[0, 50\%[$ than the fixed SAS andness of 50%. While SMS cannot directly model substitutability, the De Morgan dual of SMS can be used as a model of fixed hard substitutability (however, no soft substitutability and no adjustable orness).

(VI) Impossibility to Model Asymmetric Logic Relationships

SMS models are not able to express asymmetric logic relationships, such as a combination of mandatory, desired, and optional features, or a combination of sufficient and optional conditions.

(VII) Inappropriate Optimum Systems

The optimum systems in the case of SMS models require that for any available financial resources we must invest in each and every system component, including those that are the least significant. This is generally not realistic, particularly in cases where the resources are seriously limited.

(VIII) Inability to Model Preferential Independence

In all cases where the criterion is neither conjunctive nor disjunctive (like in the case of student GPA), we have justified cases of preferential independence and the need to use additive models. SMS cannot be used in such cases.

The SMS model can be valuable in the area of evaluation if it is used to define the average suitability ratio of two systems, *A* and *B* [BRI22, TRI00, YOO95]. If S_{Ai} and S_{Bi}, $i = 1, \ldots, n$ denote attribute scores, or attribute values, and if both scores and values are defined so that higher value denotes better system, then the ratio S_{Ai}/S_{Bi} denotes how much system A outperforms system B in the case of input i. The average ratio of systems A and B is then defined as follows:

$$R_{AB} = \prod_{i=1}^{n} (S_{Ai}/S_{Bi})^{W_i}, \ W_1 + \cdots + W_n = 1, \quad W_i > 0, \quad i = 1, \ldots, n.$$

If $R_{AB} > 1$, then R_{AB} denotes how much A is better than B. This model can be used in cases where there is no need for flexible adjustment of logic conditions.

1.3.3 Logic Unsuitability of Scoring and Utility Theory Models in Professional Evaluation

In professional evaluation, we use logic conditions to create quantitative evaluation criteria. These criteria are used to evaluate expected overall suitability of one or more competitive complex objects or alternatives and to select the best option. Evaluation and selection process is performed before the acquisition and use of the selected object. The goal of evaluation is to correctly predict satisfaction derived by stakeholder from using each of competitive objects.

The arithmetic mean is a frequent and useful component in almost all complex evaluation criteria. However, if it is used as the only component, then SAS takes the role of a general evaluation method [MIL66, MIL70, EDW82]. This is where SAS, SMS, and utility theory models fail the test of compatibility with properties of human reasoning and consequently fail to satisfy the requirements that we consider fundamental for professional evaluation.

Fixed andness decision models (SAS, SMS, and others) have very strong presence in the utility theory, which is primarily used in economics, management sciences, and related disciplines [FIS70, KEE76, KEE92, YOO95, OLS96, MOL97, KIR97, BOU00, TRI00, BEL02, KAH08, TZE11, ISH13, HEN15]. Some of listed references provide critical remarks about simple scoring but never move very far in that direction. Both additive aggregation and fixed multiplicative aggregation may occur in the context of certainty or uncertainty, crisp or fuzzy scoring, but in all cases, there is no support for flexible logic conditions. These references report many cases of the use of additive models in the area of sensitive professional evaluation. For example, those include the location or airports, selection of nuclear dump sites, and prioritizing factory automation investments [KEE76, OLS96]. According to [KOC14], the Ministry of Science and Higher Education of Poland currently uses SAS to combine four criteria (scientific and/or creative achievements, scientific potentiality, tangible benefits, and intangible benefits of the scientific activity) for evaluation of research entities in Poland.

The absence of logic conditions is particularly surprising in utility theory because utility is supposedly a psychological phenomenon where the goal is to determine the satisfying power of goods or services, and utility theory was developed "to explain human behavior in decision making" [OLS96]. This definition should yield interest in logic properties of human decision making and not in focusing on preferential independence and belief that the utility of a whole can be expressed as the sum of utilities of its parts. The use of simple additive or multiplicative models is typically justified in a scientific way by an axiomatic approach [FIS70], or in a naïve way by the attractiveness of simplicity [EDW82]. Some authors in the utility area noted that assumptions of axiomatic utility theory may not be acceptable in all cases, but the criticism of most authors rarely goes beyond that benevolent note.

To make this situation more delicate, the utility theory suffers from conflicting standpoints of cardinalists and ordinalists [OLS96, HEN15]. The classic cardinal utility theory is based on the assumption that the degree of satisfaction by goods and services is measurable and can be expressed numerically using the elusive units called *utils.* On the other hand, the ordinal utility theory believes that humans can do pairwise comparisons and produce rankings but cannot express degrees of satisfaction in absolute numerical terms [HEN15]. The strange conclusion of economists is that these two approaches are conflicting and mutually exclusive, while complex evaluation studies frequently show that cardinal and ordinal approach are complementary, cooperative, and can be simultaneously used. For example, whoever evaluates and compares mainframe computers or advanced servers knows that memory and disk storage can be easily evaluated in cardinal terms, while poorly predictable timing of some benchmark programs can be efficiently evaluated in ordinal terms; subsequently, cardinal and ordinal criteria are logically combined to yield the overall suitability of evaluated systems. In addition, practical evaluation studies show that whenever an expert provides ranking, i.e., expresses the percept that A is better or more preferred to B, the same expert is always capable of answering the natural subsequent question, to quantify to what extent is A better than B using any form of verbal or numerical rating scale.

We reject the strict ordinal approach where each pairwise comparison generates only 1 bit of information because we know that all domain experts have percepts that in pairwise comparisons can produce more than 1 bit of information. In most cases, if a decision maker cannot provide more than 1 bit in pairwise comparisons, such a decision maker does not qualify to be a domain expert and should not be given opportunities to provide pairwise comparisons.

Rigid strictly additive or equally rigid strictly multiplicative decision models, either crisp or fuzzy, either under certainty or under uncertainty, always automatically deny and reject the use of logic. This is not acceptable. Both strictly additive and strictly multiplicative models prevent the use of any form of adjustable simultaneity and substitutability in decision criteria, i.e., prevent the use of logic in decision models. Human evaluation reasoning provably requires a flexible adjustment of the degree of andness/orness in logic aggregation structures of complex criterion functions. Thus, all aggregators that use *fixed andness/orness* cannot be acceptable general models of human evaluation reasoning. These include all fixed means, SAS, SMS, additive and multiplicative utility models, and various fixed nonlinear preference and value models.

Generally, all mathematical models for professional evaluation of expected suitability of complex alternatives must be able to provide an adjustable level of penalizing systems with poor performance of any vital system component. Similarly, they must also be able to model alternative (substitutable) components and provide an adjustable level of rewarding systems that have a sufficiently high quality of any of alternative components. In addition, system evaluation models must be able to model asymmetric and compound logic relationships. SAS, SMS, cardinal, and ordinal utility models cannot satisfy these requirements. In a general case, simple scoring and utility methods are not able to model many observable and essential properties of human evaluation reasoning. In the majority of cases where SAS or SMS models are applied for solving professional evaluation and ranking problems, it is reasonable to suspect that the mathematical model is an oversimplification and that the results are questionable.

If SAS and SMS are applied for modeling the percepts of value resulting from human reasoning, then it is very naïve to expect that the subtleties of human reasoning and decision making could be reduced to simple weighted addition or equally simple weighted multiplication. This is an obvious case where we must distrust simplicity. For SAS all inputs are always optional, and for SMS all inputs are always mandatory. It is not that way in human reasoning. For both SAS, SMS, and utility models, the andness is not adjustable. Andness is definitely not constant in human reasoning. Indeed, SAS and SMS are models of only *two simple special cases* among many observable forms of human logic reasoning. Such special cases are useful as components of complex criteria, but they obviously cannot substitute a necessary general methodology that has expressive power to model a complete spectrum of observable forms of human evaluation reasoning. Elimination of drawbacks of fixed andness models is the primary motivation for development of graded evaluation logic and the LSP method presented in this book.