[16] W. S. Kim and A. K. Bejczy, "Force-reflection and shared compliant control in operating telemanipulators with time delay," *IEEE Trans. Robot. Automat.,* vol. 8, pp. 176–185, 1992.
[17] E. Paljug, T. Ohm and S. Hayati, "The JPL serpentine robot: A 12 DOF system for inspection," in *Proc. IEEE Int. Conf. Robotics Automation,* 1995, pp. 3143–3148.
[18] H. Yuasa and M. Ito, "Coordination of many oscillators and generation of locomotory patterns," *Biol. Cybern.,* vol. 63, pp. 177–184, 1990.
[19] H. Ihara and K. Mori, "Autonomous decentralized computer control systems," *IEEE Comput.,* vol. 17, pp. 55–66, 1984.
[20] S. Kimura, M. Yano, and H. Shimizu, "A self-organizing model of walking patterns of insect," *Biol. Cybern.,* vol. 69, pp. 183–193, 1993.
[21] ——, "A self-organizing model of walking patterns of insect: The loading effect and leg amputation," *Biol. Cybern.,* vol. 70, pp. 505–512, 1994.
[22] R. A. Brooks, "A robot that walks; emergent behaviors from a carefully evolved network," *Neural Comput.,* vol. 1, pp. 253–262, 1989.
[23] J. M. Cameron, D. C. MacKenzie, K. R. Ward, R. C. Arkin, and W. J. Book, "Reactive control for mobile manipulation," in *Proc. IEEE Int. Conf. Robotics Automation,* 1993, pp. 228–235.
[24] D. W. Payton, D. Keirsey, D. M. Kimble, J. Krozel, and K. Rosenblatt, "Do whatever works: A robust approach to fault-tolerant autonomous control," *J. Appl. Intell.,* vol. 2, pp. 225–250, 1992.

The Measurement of a Design Structural and Functional Complexity

Dan Braha and Oded Maimon

***Abstract*— The complexity of a design process or a design artifact substantially influences their performance. When evaluation of terms such as "design complexity" and its "quality" is addressed in studies, it is often performed in an *ad hoc* manner. This paper attempts to remedy this situation by articulating two definitions of design complexity (*structural* complexity versus *functional* complexity), their associated value measures, and the relationships between them. The *structural* definition states that a design complexity is a function of its representation. Defining design complexity in the structural way provides quantitative techniques for evaluating vague terms such as "abstraction level," "design form's size," and "designing effort." The *functional* definition states that a design complexity is a function of its probability of successfully achieving the required specifications (functional requirements and constraints). The proposed measurable metrics provide a proper basis for evaluating each step of the design process, and accordingly recommends the direction to follow for design modification and enhancement. It also provides a framework for comparing competing artifacts (the output of a design process). The paper concludes by discussing the scope of the measures.**

I. Introduction

A. Complexity Judgment of Artifacts and Design Processes

The study of design complexity is motivated by several reasons: 1) design complexity valuation method can support the evaluation of artifacts developed in research or practice, and the determination of their relative merit. This evaluation is essential for providing feedback on research progress, such as when developing products by prototyping; 2) design complexity valuation methods can help identify good information to be used by designers or for incorporation in computer aided design systems; and 3) when an intelligent computer aided design system has several competing modules (e.g., production rules) for solving a design task, design complexity valuation methods can identify which module to invoke for achieving the task.

In order to maintain the focus on the methodological aspects of complexity evaluation, the discussion in this paper will refer to design complexity without reference to human design. By this we limit the discussion to the valuation of complexity embedded in intelligent computer aided design systems.

Manuscript received February 26, 1995; revised July 5, 1997 and February 28, 1998.

D. Braha is with the Department of Industrial Engineering, Ben-Gurion University, Beer-Sheva 84105, Israel.

O. Maimon is with the Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel.

Publisher Item Identifier S 1083-4427(98)04356-2.

II. Artifact Complexity

Artifact's complexity may be said to be the exact converse of simplicity. It is generally acknowledged that the lower the artifact's complexity, *ipso facto*, the greater the artifact's simplicity, whereas concentration on simplicity leads to enhanced artifact's reliability and quality at lowest cost [1]–[3]. Simplicity is a concept used by designers and aestheticians for many centuries and remains a principle of considerable concern to them. By applying the word "simple" to a work of art, the designer is not suggesting that the work is deficient. Simple means that the design is being reduced to the fewest possible lines, shapes and subparts, without reducing its functional requirements or violating its specifications. Thus, simple design will reduce the assembly time and product cost, as well as increase reliability by many orders of magnitude. The simplicity principle implies that if a design satisfies more than the minimum number and measure of functional requirements originally imposed, the part or process may be over-designed. Moreover, the elimination of functional requirements originally imposed will cause an incomplete design. In other words, "good" designs must be complete, yet not burdened with nonessential details. Carrol and Bellinger [1] remark that the superior design is one which encompasses the necessary operating and protective functions with the absolute minimum number of components and relations. Suh's axiomatic theory of design [4] also provides overwhelming evidence through numerous principles that support the "simplicity" contention (e.g., "minimize the number and complexity of part surfaces for greater efficiency"). Pugh [5] remarks that the theme of simplicity may have emerged from endeavors to avoid discontinuities in systems. A discontinuity may be a bend or fork in the road, or a constriction in a pipe. Mahmoud and Pugh [6] offer a costing method for turned components produced by a variety of machines. Their costing equation is based on the number of discontinuities in the system, subsystem or component. Thus, it may be inferred that the more discontinuities you have in a system, subsystem or component, the more unreliable it will tend to be. Moreover, the number of discontinuities effects the cost due to excessive processing operations, which also has a significant impact on quality and reliability.

In software engineering, artifact descriptions range from formal languages (e.g., symbolic programming and hardware design/description languages) to very informal and visual descriptions (such as functional block diagramming and flow-diagrams). The increasing complexity of computer programs has increased the need for objective measurements of software complexity. The major ways of measuring *software complexity* in quantitative terms fit into four categories: 1) *structured-based* measures that look to the pattern of the control flow in the software [19]; 2) *feature-based* measures that look to a selection of characteristics visible from the documentation of either the design or the source code [20]; 3) *function-based* measures that look to the pattern of input to output data correspondences [21]; and 4) *token-based* measures that look to the manner of expression of the software [22]. The *structural* complexity measures presented in Sections IV–IX are based on the token-based measures of software complexity as given in [22].

III. Design Process Complexity

The design process was defined in [15]–[17] as an iterative scheme of decomposition and mapping between the functional and artifact domains. This iterative scheme at each level of decomposition and mapping is not unique. The evolution at each level depends on the imagination and experience of the designer, and there is no guarantee whatsoever that the same process would unfold. Thus, it is desirable to develop a systematic and generalizable framework for design process complexity valuation. Design process complexity valuation measures can aid the designer in evaluation and comparing decomposition and mapping alternatives at each level of the design process hierarchy. For example, design complexity is addressed in Suh's axiomatic design [4] by the Independence axiom: in an acceptable design, the mapping between the functional requirements and design parameters is such that each functional requirement can be satisfied without affecting any other functional requirements. At each level of decomposition and mapping, the Independence axiom is used to distinguish between acceptable and unacceptable solutions. The simplest type of design process that satisfies the Independence axiom at each level of the design process hierarchy is an uncoupled design process. In an uncoupled design process, each functional requirements can be changed without affecting any other functional requirement, which means that each functional requirement is ultimately controlled by a unique design parameter. Coupled design processes are considered highly complex and the designer should consider other mapping alternatives.

A. Two Definitions of Design Complexity

Before discussing how design complexity may be measured, we articulate two definitions of design complexity. By "design complexity" we mean either artifact complexity or design process complexity. Both definitions assume that the study of design complexity is related to the *valuation of information* embedded in the design as captured, for example, by intelligent computer aided design systems. The valuation of information depends on what we define as information. The following definitions of design complexity dictate the type of valuation measures that can be applied:

1) Structural Design Complexity: Design complexity is a function of the design's *information content.* Information is *whatever is represented*. The design's representation may include facts, causal relations, mathematical models, etc. Defining information in the *structural* way states that the "quantity" of information may be measured directly based on its internal structure. Design complexity is therefore a static entity.

The structural definition has several appealing properties: 1) it facilitates easy valuation of design complexity performed by simply inspecting the declarative evolutionary structure of design (with or without considering the inference mechanisms)—this is much cheaper than executing behavior assessment experiments—and 2) two "amounts" of information (e.g., facts, rules, and laws included in the designer's knowledge body) could be added to yield a larger knowledge body, or knowledge could be transferred between intelligent computer aided design systems.

There are some limitations with the structural definition: 1) it detaches design from its ultimate purpose of satisfying initial specifications, therefore, the purpose of design is not tested and, at best, can only be hypothesized from the structural measure; and 2) it detaches information from its method of acquisition. Consequently, when information acquisition terminates, some meaning may be unrecoverable, and 3) it cannot explain actions that are not logical reasoning, inconsistencies, and psychological phenomena.

Applying the structural definition to evaluating artifact complexity means that if two artifacts (as captured, for example, by computer aided design databases) satisfy the required specifications, the best artifact (in terms of design complexity) is the one with the minimum information content. Thus, the complexity of an artifact may be said to be a function of its information content.

Defining design process complexity in the structural way means that if two design processes successfully achieve the required specifications, the best design process (in terms of design complexity) is the one with the minimum total information content. Thus, the complexity of a design process may be said to be a function of its information content at each level of the design process hierarchy.

There is also a definition of information as a dynamic entity.

2) Functional Design Complexity: Defining information in the functional way states that "information" is a distinct notion, independent of representation. Representations exist at the symbol level and not at the information level. In addition, information serves as the specification of what a symbol structure should be able to do. Therefore, information has a purpose, and is what a design has that allows it to attain goals.

Defining design process complexity in the functional way means that if a designer has information that one of its decisions will lead to one of its goals, than the designer will select that decision (the "principle of rationality"). Therefore, information manifest and should be evaluated functionally. This means that information can be described in terms of its operation to satisfy the goals of the system. Alternatively, two design processes may be compared based on their output. The best design process (in terms of design complexity) is the one that yields an artifact in which its probability of successfully achieving the required specifications is maximized.

A similar "principle of rationality" is applied when evaluating artifact complexity in the functional way. Often, the behavior of an artifact is nondeterministic. Functional requirements are therefore satisfied only to a degree. Thus, a design complexity may be said to be a function of its probability of successfully achieving the required specifications. In all cases, the best artifact is the one that maximizes the probability of successfully achieving the required specifications.

B. Organization of the Paper

The rest of the paper is organized as follows. Applying the token-based measures of software complexity as given in [22], Sections IV–IX present structural design complexity measures in view of the evolutionary design process developed in [15]. First, some basic metrics are introduced. Second, we use the basic metrics to derive algebraic formulas for the information content of design and related *structural* complexity measures. Section III shows that the proposed complexity measures also lead to the ability to rapidly estimate the approximate total assembly time of a product, assembly efficiency, and product defect rates. Section IV explores the relationships between the theoretical approach used to derive the structural complexity measures and thermodynamics. Section V presents a *functional* design complexity. Section VI concludes the paper.

IV. Structural Design Complexity Measures

A. Description of the Valuation Measures

In this section, we develop quantitative metrics for measuring structural design complexity. The proposed complexity measures can be used by the designer at any stage of the design process, from the conceptual design to the finality of detailed design. Following [15] and [16], the term "design" is defined as an evolutionary process. Given a description of a desired function and constraints, called specifications, the designer (or intelligent computer aided design system) provides a representation of an artifact that produces the function and satisfies the constraints. This representation, called an artifact description or artifact structure, was identified in [15] as an algebraic structure. By adaptively modifying the *design form* (the conjunction of pair of tentative artifact and postulated specifications), from one step to the next, the designer arrives at a design solution. A design form at any abstraction level (synthesis stage) is a description of first-order and high-order constraints which a physically implemented artifact should satisfy. Following the foregoing discussion, we define design process complexity to be a function of the "quantities" of *information content* (also termed "*size*") embedded in each level of the design process. The artifact complexity in each level of the design process may be evaluated by applying the complexity measures to the tentative artifact part. The same approach holds for the successful artifact realized in the lowest level in the design process hierarchy (Level 1), where the artifact is described in detail. The information content has yet to be modeled. Defining information content in the structural way means that the information content should be a function of the description or representation of the design form in some *symbolic medium*, e.g., diagrams, computer languages, or first-order logic. Therefore, it may be practical to consider the design form as a computer program, i.e., an ordered string of *operators* and *operands* [22]. An operand is a variable or a constant and an operator is an entity that can alter either the value of an operand or the order in which it is altered. For example, the artifact representation model presented in [15] is similar to the machine language of the computer, in which each instruction contains an operation code (a relation) in one segment and an operand or the address of an operand (an assignment) in the remaining section of the instruction. Henceforth we will use "operators" and "operands" as generic terms which include "relations" and "modules" as a special case. No other category of entities need be present.

We expect the *information content* of the tentative design form to vary in the course of the design process. Other "quantities" that vary in the course of the design process are the *abstraction level*, as well as the designer *effort* at arriving at the tentative design form. The aims of this section are twofold: 1) to provide quantitative techniques for describing and evaluating such qualitative terms as "information content," "abstraction level," and "designing effort" and 2) to introduce the time complexity measure, which may be used to rapidly estimate the approximate total assembly time of a product, assembly efficiency, and product defect rates.

B. Basic Measures

Given a description of the design form in some symbolic medium (e.g., first-order logic) at a particular abstraction level, it is possible to identify all the operands and operators that the artifact representation employs (supplemented with such distinguished operators as "$\sim$," "$\wedge$," "$\vee$," "()," etc.). Similarly, it is also possible to identify all the operators and operands that constitute the specification part. Let the finite set of operators and operands (the *alphabet*) be denoted by Ω. From these simple definitions, it is possible to obtain quantitative measures for many useful properties of design processes.

Since a design form consists of an ordered string of operators and operands, and nothing else, it can be characterized by basic measures that are capable of being counted or measured as follows:

- ρ number of unique or distinct operators appearing in the design form;
- N number of unique or distinct basic operands appearing in the design form;
- N_1 total number of occurrences of the operators in the design form;
- N_2 total number of occurrences of the operands in the design form.

The size of the alphabet is defined to be

$$\eta = \rho + N \tag{1}$$

and the length of the design form to be

$$L = N_1 + N_2. \tag{2}$$

Example 1: In the final stage of design of a serial binary adder unit, the design process terminates with the following successful electronic circuit. To characterize the basic measures associated with the electronic circuit, a description of the electronic circuit in some symbolic medium needs to be considered. Here, we use the following representation in first-order predicate calculus:

$$\text{AND}(A,\, S_2,\, S_3) \wedge \text{AND}(C,\, B,\, S_1) \wedge \text{OR}(B,\, C,\, S_2) \wedge \text{OR}(S_1,\, S_3,\, \text{Output}).$$

Thus, the following basic measures (constituting the alphabet) are identified: 1) signals are classified as operands, 2) logical devices are classified as operators, and 3) the distinguished symbols "$\wedge$" and "()" are classified as operators. From the counts of operators and operands we see that

$$(\rho,\, N_1) = (4,\, 12) \ \ (N,\, N_2) = (7,\, 11).$$

C. Composite Measures

Based on the basic measures presented above, we present in the remaining section three measures (based on [22]) that may be used to express the complexity (the converse of simplicity) inherent in the design process. The proposed measures reflect such qualitative terms as *information content, abstraction level,* and *designing effort.*

V. Information Content

As mentioned earlier, the design process of any complex system proceeds on a stage-by-stage basis, and the description at each stage denotes the design form (i.e., a pair of tentative $\langle$artifact, specifications$\rangle$) of the system at a particular *level of abstraction.* The characteristics of the various abstraction levels depend on the designer and the problem domain. The design form at level i may be viewed as a decomposition of the design form at level $i+1$.

An important characteristic of a design form at a particular abstraction level is its *size*. Whenever a given design is translated from one abstraction level to another, its size changes. To study such changes in a quantitative way requires that the size be measurable.

A suitable metric for the size of any design form, called the *structural information content* H, can be defined as

$$H = L \log_2 \eta = (N_1 + N_2) \log_2 (\rho + N). \tag{3}$$

The quantitative concept of the information content of a design form can be interpreted as follows: since the number of different entities in any design form is given by its alphabet, it follows that the number of bits required to provide a unique pointer (designator) for each entity must be given by $\log_2 \eta$. Using this number of bits to specify each item in the string of length L gives the information content. Thus, the information content can be interpreted as the fewest number of binary bits with which it could be represented. Later, when defining the designing effort, we will provide a more in-depth analysis of the design information content.

VI. Information Content and Entropy

We now consider the relationship between the information content of a design form and its entropy. Following the seminal work by Shannon [7] and Wiener [8], information is defined as a reduction in uncertainty. The uncertainty preceding the occurrence of an event is usually termed *entropy.* Information theory provides a way of quantifying the *information content* received, so that the quantity of information received is equal to the reduction in entropy.

Let $X_1, X_2, \cdots, X_L$ be independent, identically distributed (i.i.d.) discrete random variables drawn according to the probability mass function $p(x) = 1/|\Omega| = 1/\eta$, where each variable X_i is either an operator or an operand over the alphabet Ω (the finite set of operators and operands). The probability of a sequence $(x_1, x_2, \cdots, x_L) \in \Omega^L$ is $1/|\Omega|^L = 1/\eta^L$. According to information theory, the *information content* received from a *particular* design form with length L is equal to the *joint entropy* $H(X_1, X_2, \cdots, X_L)$. The joint entropy of a sequence of discrete random variables $(X_1, X_2, \cdots, X_L)$ is defined as

$$H(X_1, X_2, \cdots, X_L) = -\sum_{x\in\Omega^L} p(x_1, x_2, \cdots, x_L) \cdot \log_2 p(x_1, x_2, \cdots, x_L).$$

Since the probability of a sequence $(x_1, x_2, \cdots, x_L) \in \Omega^L$ is $1/|\Omega|^L = 1/\eta^L$, it is easy to see that the information content associated with the particular design form is $H(X_1, X_2, \cdots, X_L) = L \log_2 \eta$, in accordance with the foregoing definition.

VII. Minimal Information Content

In translating from the higher abstract design form to the lower abstract level, we use a greater number of simpler operators and operands.

Expressing the design form, even in the highest abstraction level, would still require operators and operands. The highest (most compact) design level is formulated in ambiguous or imprecise terms. Each operator in the most compact version of the design form represents a distinct functional requirement that the required design solution is expected to satisfy. The number of operands in the most compact design form would depend upon the design problem itself, and would equal to the number of conceptually unique input and output operands of each functional requirement (see [22]).

Denoting the corresponding parameters in a design's most compact representation by asterisks, it follows from (3) that the minimal (also initial) information content is given by

$$H^* = L^* \log_2 \eta^*. \tag{4}$$

Note that when the initial specification includes a single functional requirement θ^0, the design's most compact representation may be given by $\theta^0(i_1, i_2, \cdots, i_{N^*})$, where $i_1, i_2, \cdots, i_{N^*}$ denote the unique input and output operands. Substituting $L^* = \eta^*$ and $\rho^* = 2$ (considering the operators "θ^0" and "()"), we obtain

$$H^* = (2 + N^*) \log_2(2 + N^*). \tag{5}$$

Example 2: Let the initial design form (Step 1), in the binary-adder design process, be the following specification SUM(X, Y, carry-in, sum) ∧ CARRY(X, Y, carry-in, carry-out). Hence, the initial design form consists of four operators ("SUM," "CARRY," "∧," and "()") and five operands ("X," "Y," "carry-in," "carry-out," and "sum"), thus: $H^* = 13 \log_2 9 = 41.2$ bits.

Since the minimal information content evaluates the initial design form, it may also be termed as the initial information content. Consequently, the minimal information content of a design is a single-valued function of N^* (the number of conceptually unique input or output operands). The minimal information content is the minimum possible information content associated with any given design problem in the course of the design process. It represents an absolute value against which other information contents, which are evaluated in subsequent stages, can be compared. Thus, it may be considered to be a general measure of the content of any design problem. Furthermore, it follows that H^*, unlike H, must be completely independent of the technology (e.g., methodology, design paradigm, or computerized tools) or the abstraction level in which the detailed artifact is expressed.

VIII. Abstraction Level

As mentioned earlier, the level of abstraction has gone down as the information content has gone up. The information content of a given design form (at a particular abstraction level) must inversely reflect the level at which it is implemented, and the ratio of the information contents of two design forms (for the same design problem) must give the inverse of the ratio of the levels at which they have been implemented. This leads to the definition of abstraction level A as

$$A = \frac{H^*}{H}. \tag{6}$$

In the initial synthesis stage, the design form $H^* = H$; therefore, $A = 1$ (the highest level). In a lower abstraction level, the design form would have more operators and operands. Therefore, as expected, $A = H^*/H$ is less than 1 at the lower level. Furthermore, the product of information content times level will be completely abstraction level independent, because $H^* = A \cdot H$.

Example 3: Let us compute the artifact abstraction level associated with the electronic circuit of Example 1. As shown in Example 1, the basic measures are $(\rho, N_1) = (4, 12)$, $(N, N_2) = (7, 11)$. Thus the information content associated with the electrical circuit is: $H = (N_1 + N_2) \log_2(\rho + N) = 23 \log_2(11) = 79.6$ bits. The minimal information content associated with the electrical circuit (as shown in Example 2) is: $H^* = 13 \log_2 9 = 41.2$ bits. Therefore, the abstraction level as defined in (6) is: $41.2/79.6 \approx 0.51$.

IX. The Designing Effort and Time

The designing effort provides a measure for the "mental" activity required to reduce a design problem (expressed by means of initial goals) to an actual abstraction level. The foregoing metrics and concepts provide a useful frame of reference for its quantification. The construction of a design form consists of the judicious selection of L entities (operators and operands) from a list of η entities. If a binary search method is used to select entities from the alphabet of size η, then on the average the total number of mental comparisons needed to construct a design form, is the same as the previously defined information content $H = L \log_2 \eta$. Expressing the number of "elementary mental discriminations" [9] required to make one average mental comparison as $1/A$, gives the effort E in units of elementary mental discriminations

$$E = \frac{1}{A} \cdot H. \tag{7}$$

TABLE I
MEASURES FOR THE INITIAL DESIGN FORM OF THE FASTENER DESIGN PRESENTED IN [15]

Design Form 0	Measures
HIGH-STRENGTH $\wedge$ HIGH-RETRACTABILITY $\wedge$ MEDIUM-PRECISION	$\rho^* = 1$ $N^* = 3$ $\eta^* = 4$ $N_1 = 2$ $N_2 = 3$ $L^* = 5$ $H^* = 10$ $H = 10$ $A = 1$ $E = 10$ $T = 0.55$ (S=18)

TABLE II
COMPLEXITY MEASURES FOR THE FASTENER DESIGN PROCESS PRESENTED IN [15]

	ρ	N	η	N_1	N_2	L	H^*	H	A	E	T
1	1	3	4	2	3	5	10	10	1	10	0.55
2	1	4	5	3	4	7	10	16.25	0.61	26.4	1.46
3	1	5	6	4	5	9	10	23.26	0.42	54.1	3.00
4	1	5	6	4	5	9	10	23.26	0.42	54.1	3.00
5	1	5	6	4	5	9	10	23.26	0.42	54.1	3.00
6	1	6	7	5	6	11	10	30.88	0.32	95.35	5.29
7	1	6	7	5	6	11	10	30.88	0.32	95.35	5.29
8	1	4	5	3	4	7	10	16.25	0.61	26.24	1.46
9	1	5	6	4	5	9	10	23.26	0.42	54.10	3.00

Note that the *effort complexity measure* E is a linear function of the information content H (with the constant coefficient $1/A$). E can be also used to estimate the amount of effort required by an experienced designer to understand or comprehend the design form.

Equation (7) can be converted directly into units of time, merely by knowing the rate, S, at which the brain makes elementary mental discriminations [9]–[11], [22]. Provided the designer is concentrating, experienced, and has a complete specification of the design problem, the relation between time and effort is expected to be

$$T = \left(\frac{1}{S \cdot A}\right) \cdot H = \frac{H^2}{H^* S}. \tag{8}$$

Note that the *time complexity measure* T is a linear function of the information content H (with the constant coefficient $1/(S \cdot A)$). The time T may also be interpreted as the time used to comprehend a design form (by reading it).

Example 4 (Quantitative Analysis of a Design Process): As mentioned earlier, the measurable metrics developed in this paper may be estimated for all design forms that are encountered in the course of the design process. Thus, in working the design process, these measures will always be visible and can be continuously monitored. We shall demonstrate how this approach maps onto the evolutionary design of the mechanical fasteners presented in [15]. We shall provide a self contained description of the evolutionary structure of the complexity measures underlying the design process. The initial design form (artifact and specification parts) and its complexity measures are presented in Table I. Table II depicts the measures for the whole design process.

X. EVALUATING THE TOTAL ASSEMBLY TIME OF A PRODUCT

A. Total Assembly Time and Time Assembly Measure

The complexity of assembly of a product can be gauged by the time required to perform the assembly. In this section, we show that the time complexity measure T may be used to rapidly estimate the approximate total assembly time of a product. This provides a powerful analytical tool that is useful during concept development.

Following the *structural* definition of complexity, the complexity of assembly of a product is a function of its representation. In the sequel, it is suggested to use a representation that embeds the information associated with the assembly interfaces. We focus on the assembly interfaces for the following reason. As the number of mating features in an interface increases, there are additional restrictions on the orientation of the parts during assembly. As a result, the assembly time increases as the complexity of the interface grows. For example, a part with only one correct alignment orientation must have more interface dimensions and a longer assembly time than a simple cylinder that can be inserted in either axial direction into a hole.

Based on the foregoing assumption, we use the following representation in first-order predicate calculus:

$$\text{INTERFACE}(m_{i_1}, m_{i_2}) \wedge \text{INTERFACE}(m_{i_3}, m_{i_4}) \wedge \cdots \wedge \text{INTERFACE}(m_{i_{k-1}}, m_{i_k})$$

where the operands m_{i_j} denote the parts of the assembly, and the operator "INTERFACE" represents the liaisons between two separated parts. The information content H, associated with this representation, can then be computed. The most compact representation associated with the product assembly may be given as follows:

$$\text{INTERFACE}(m_1, m_2, \cdots, m_N)$$

where m_i denote the parts of the assembly. The minimal information content H^* can then be computed. Finally, the approximate total assembly time of a product as given by (8) is: $T = H^2/H^*S$, where S varies between 5 and 20 (we often use $S = 18$).

In [17], we inspect through an extensive statistical analysis the correlation between the time complexity measure T and the estimates of product assembly times that were derived by Boothroyd and Dewhurst in their Design for Assembly (DFA) structured methodology [12]. The correlation between the time complexity measure T and the Boothroyd and Dewhurst's estimates is found to be very close to ±1 over a wide diversity of experiments. This demonstrates that the time complexity measure T may be used as a powerful predictive tool. By simply determining the number of interfaces and number of parts in each product concept, the approximate total assembly time can be determined with a minimum amount of analysis and without any dependence on a database. Such a tool could be used in the earliest stages of concept development to estimate the approximate total assembly times, allowing comparison of competing concepts or stimulating redesign at the time when it is easiest to make design changes.

The time complexity measure T reveals two fundamental factors that can contribute to assembly time: 1) the number of assembly operations (a subset of the set of assembly interfaces) and 2) the number of parts (a subset of the set of assembly operations). While the role of part count has long been recognized as the measure of design effectiveness [13], the method described here provides a critical missing link that relates the product assembly time to the number of operations. Without these relationships, it is impossible to accurately compare concepts that differ in the number of parts and operations.

B. Assembly Defect Rates and Time Assembly Measure

Even when parts satisfy defined tolerances and requirements, defects can occur during the assembly process. One source of assembly defects is *interference* between mating parts. An evaluation of several simple assemblies demonstrated that this will lead to an increased probability of assembly interference due to variations in dimensions, even for parts toleranced by the best current methods. Thus, as previously mentioned, *the assembly time increases as the complexity of the interface grows* [23].

Assembly errors, such as installing a part in an incorrect position or orientation are other sources of assembly defects, which can occur during assembly even in the case of perfect parts and minimum complexity interfaces (e.g., a part may be omitted). For example, misalignment during insertion can damage mating parts that are otherwise functionally adequate. As the difficulty of the task increases, the probability of an assembly error is also likely to increase for the same level of care in the operation. Each increase in assembly time can be related to an increase in the difficulty of the assembly operation. *Thus, the probability of an assembly error should also be a function of the assembly operation time* [23]. The finding that defects would increase with total assembly time, combined with the potential of applying the time complexity measure T to rapidly estimate the approximate total assembly time of a product [17], provides a method of rapidly estimating product defect rates.

C. Design Assembly Efficiency and Time Assembly Measure

Practitioners tend to focus on part count as the measure of design effectiveness [13]. However, as mentioned above, part count is an inadequate and potentially dangerous focus for design.

The search for a better criterion led to a study of Assembly Efficiency, a parameter introduced by Boothroyd and Dewhurst in their DFA structured methodology [12]. Assembly efficiency compares computed assembly times to an ideal but arbitrary standard. This relationship is expressed as follows [12], [23]:

$$EM = \frac{t_{\text{ideal}} \cdot NM}{TM} \tag{9}$$

where

- EM manual assembly efficiency;
- t_{ideal} "ideal" assembly time per part, suggested as 3 s [12];
- NM theoretical minimum number of parts, determined as the number of parts that satisfy at least one of the following three criteria: 1) must move during operation, 2) must be made of different material, and 3) must be separate to permit assembly or disassembly;
- TM total manual assembly time in seconds.

The assembly efficiency parameter can be interpreted as a measure of the potential to achieve further reduction in assembly time by redesign. The importance of the assembly efficiency parameter is recently being acknowledged. A significant relationship was observed between the defect rate in the factory assembly of several mass produced electromechanical products and the assembly efficiency parameter [23].

Using the time complexity measure T as an estimate of total assembly time, combined with (9), we define the *assembly efficiency measure* in terms of the theoretical minimum number of parts, and the time complexity measure T

$$\tilde{EM} = \frac{3 \cdot NM}{T}. \tag{10}$$

The strong linear correlation between the time complexity measure T and the estimates of product assembly times that were derived by Boothroyd and Dewhurst (see [17]) also supports the assumption that the manual assembly efficiency increases with the assembly efficiency measure. Considering that much less information is needed to derive the assembly efficiency measure, it makes it the more elegant and simple method.

XI. Thermodynamics and the Design Process

This section explores the analogy between the foregoing design complexity measures and thermodynamics. This analogy also enables to generalize the proposed complexity measures.

In order to demonstrate the analogy between a general thermodynamic process and a design process, we must first state definitely what the *system* is and what the *environment* is. In thermodynamics, the system interacts with its environment through some specific thermodynamic process, starting from an initial state to a final state. During this process, energy in the form of heat (Q) and work (W) may go into or out of the system.

TABLE III
Thermodynamics and the Design Process

Thermodynamic Process	Design Process
Balloon	Design Form
Blower + Gas	Designer
Pressure	Abstraction Level
Volume	Information Content
Internal Energy	Designing Effort
Entropy	Information Content & Length
Power	Stroud Number
Inflationary Time	Designing Time

Let us now compute Q and W for a specific thermodynamic process. Consider a gas contained in a balloon, and assume that no heat flows into or out of the system (an irreversible adiabatic process). Let the balloon be the system, and let the blower and gas represent the environment. The foregoing physical thermodynamic process corresponds to the design process as summarized in Table III. Based on this analogy, we derive the proposed design complexity measures.

Initially, the balloon is in equilibrium with the environment external to it, and has a pressure of P_i and a volume H_i. Work can be done on the balloon by compressing the gas. Consider a process whereby the system interacts with its environment and reaches a final equilibrium state characterized by a pressure P_f and a volume H_f.

The work done by the gas in displacing the balloon is given by

$$W = \int dW = \int_{H_i}^{H_f} P\, dH. \tag{11}$$

This integral can be graphically evaluated as the area under the curve in a P–H diagram.

There are many different ways in which the system can be taken from the initial volume H_i to the final volume H_f. However, from the first law of thermodynamics and the adiabatic process ($Q = 0$), we obtain

$$U_f - U_i = -W \tag{12}$$

where U_f, the internal energy of the system in state f, minus the internal energy of the system in state i, is simply the change in internal energy of the system. Moreover, this quantity has a definite value independent of how the system went from state i to f.

By analogy with the design process (cf., Table III), we consider the special case in which the initial and final volumes $H_i (= H^*)$ and H_f represent the information content [given by (3)] of the initial and terminal design form, respectively. The pressure is chosen to be $P(H) = \kappa(H/H^*)^\gamma$, assuming κ and γ are constants that characterize the design problem and its features. This type of expression is consistent with the *power law* and sizing model which is frequently used for estimating the cost of equipment [14]. Then

$$\begin{aligned} W &= -\int_{H^*}^{H_f} \kappa \left(\frac{H}{H^*}\right)^\gamma dH = \frac{\kappa H^*}{\gamma+1} - \frac{\kappa (H_f)^{\gamma+1}}{(H^*)^\gamma(\gamma+1)} \\ &= \frac{\kappa H^*}{\gamma+1} - \left[\frac{\kappa}{A(\gamma+1)}\right] \cdot H_f \end{aligned} \tag{13}$$

where A is defined as in (6).

Therefore by (12) we define the internal energy as $U(H) = [\kappa/A(\gamma+1)] \cdot H$. Note that W is negative when work is done on the system.

To illustrate a case, let $(\kappa, \gamma) = (2, 1)$. We obtain

$$P(H) = 2\left(\frac{H}{H^*}\right) = 2A^{-1} \tag{14}$$

$$W = H^* - \frac{(H_f)^2}{H^*} \quad \text{and} \quad U_f - U_i = \frac{(H_f)^2}{H^*} - H^*. \tag{15}$$

Thus $U(H) = (H)^2/H^*$, which is exactly the effort expression given by (7).

We now formulate the design process analogy of the second law of thermodynamics. The second law of thermodynamics can be stated, loosely, as: There exists a useful thermodynamic variable called *entropy* that is characteristic only of the state of the system, and an irreversible adiabatic thermodynamic process that starts in one equilibrium state and ends in another. This system will go in the direction that causes the entropy of the system to increase.

In statistical mechanics the quantitative relationship between entropy and disorder is given by the relation

$$\varrho = k_N \ln w. \tag{16}$$

Here, k_B is Boltzmann's constant, ϱ is the entropy of the system, and w is the probability that the system will exist in the state it is in relative to all the possible states it could be in. This equation connects a thermodynamics or macroscopic (entropy) quantity, with a statistical or microscopic quantity, the probability.

Let us identify the corresponding probability w for the design process case. Here, the alphabet, η [see (1)], changes in the course of the design process. The probability of finding a particular operand or operator in a given design form is

$$w^* = \frac{1}{\eta}. \tag{17}$$

Thus, assuming the operands and operators are independently chosen, the probability that a given design form may be found in a certain design stage is

$$w = \left(\frac{1}{\eta}\right)^L \tag{18}$$

where L is the length of the design form. Equation (18) coupled with (16) leads to the following entropy ("k" in this equation is analogous to Boltzmann's constant)

$$\varrho = -k \ln\left(\frac{1}{\eta}\right)L. \tag{19}$$

Since the entropy ϱ is proportional to the length and information content (i.e., $\varrho \propto L$ and $\varrho \propto H$), we may identify information content and length with the qualitative idea of disorder and entropy.

Let us now consider the time involved in doing work on the balloon if the system is taken from an initial volume $H_i = H^*$ to a final volume $H_f = H$. Define the power S as the time rate at which work is done. If the power delivered by the blower (designer) is constant, then

$$W = S \cdot T \Rightarrow T = \frac{\kappa(H_f)^{\gamma+1}}{(H^*)^\gamma(\gamma+1)S} - \frac{\kappa H^*}{(\gamma+1)S}. \tag{20}$$

Continuing the simile with the design process, we let $(\kappa, \gamma) = (2, 1)$ and let S be the Stroud number (the rate at which the brain makes elementary mental discriminations). Hence

$$T = \frac{(H)^2}{H^* S} - \frac{H^*}{S}. \tag{21}$$

T represents the marginal time involved in changing the system from an initial volume to its final volume. Thus, we conclude that the time involved in changing the system from an initial volume $H_i = 0$ to a final volume $H_f = H$ is given by

$$T + \frac{H^*}{S} = \frac{(H)^2}{H^* S}. \tag{22}$$

Equation (20) validates the time equation derived earlier for design processes (8).

To summarize, we have argued that by analogy with thermodynamics, we may develop scientific design complexity measures. By this approach, we attempt to understand—or at least quantitatively assess—the microscopic design process by applying large-scale or macroscopic formulas.

XII. Functional Design Complexity Measure

As mentioned in Section I-B, the study of design complexity is related to the *valuation of information content* embedded in the design. In Section I-B, we defined information in the functional way as the specification of what a symbol structure (e.g., an artifact or a design process) should be able to do. That is, information has a purpose, and is what a design has that allows it to attain goals.

Defining information content in the functional way means that the capabilities of each solution alternative may be compared with the governing set of requirements until the designer identifies the solution alternative that best satisfies the functional requirements. Without a numerical basis for comparison, however, the final selection of a design solution involving many functional requirements can only be made on a subjective or ad hoc basis. The ability to quantify how well a proposed artifact satisfies the governing requirements provides a rational means for selecting the best solution [4].

In this section, we define the information content of an artifact to be a function of its probability of successfully achieving the functional requirements (abbreviated as the *probability of success*). Functional information content is defined as the logarithm of the inverse of the probability of success p (see also [4])

$$F = \log_2\left(\frac{1}{p}\right). \tag{23}$$

The probability of success p that relates to the satisfaction of a given functional requirement can be computed as follows. A requisite tolerance is associated with the given requirement. The anticipated *response*, r, from a proposed artifact is represented as a probability density function $f(r)$. The probability of satisfying the functional requirement is given by the area, which falls between the limits defined by the requisite tolerance. Thus, the probability of success and functional information content are given by

$$\begin{aligned} p &= \Pr ob[a \le r \le b] \\ &= \int_a^b f(r)\,dr \Rightarrow F = \log_2\left[\frac{1}{\int_a^b f(r)\,dr}\right]. \end{aligned} \tag{24}$$

The success probability can be increased by moving the mean of the response toward the desired tolerance and then reducing its variance. In addition, while the success probability increases, the functional information content and the artifact complexity decrease.

When there are n independent functional requirements to satisfy, the overall probability of success is given by

$$p = \prod_{i=1}^{n} p_i \tag{25}$$

where p_i is the probability of satisfying the ith functional requirement as given in (24). Applying (23), the total functional information content is given by the sum of the information contents associated with each functional requirement, i.e.,

$$F = \log_2 \left(\frac{1}{\prod_{i=1}^{n} p_i} \right) = \sum_{i=1}^{n} \log_2 \left(\frac{1}{p_i} \right) = \sum_{i=1}^{n} F_i. \tag{26}$$

Let us consider the case where the anticipated *response,* r, is represented as a uniform probability density function: $f(r) = 1/(d-c)$ for $c < r < d$, and $f(r) = 0$ otherwise. The uniform probability distribution function is used in situations where the designer has no *a priori* knowledge favoring the distribution of responses except for the end points; that is, the designer does not know what the shelf life of an electrical receptacle will be but it must falls, say, between 720 and 800 h. In the case of a uniform probability distribution, it is clear from (24) that the probability of success p is equal to the ratio of the region of overlap between the design tolerance $[a,\ b]$ and the response range $[c,\ d]$. Thus, the functional information content can be simply written as

$$F = \log_2 \left(\frac{d-c}{\max(a,\ c) - \min(b,\ d)} \right). \tag{27}$$

Example 5: Consider the design of a flexible manufacturing system (a detailed example is shown in [18]), where the required functional requirement is represented in terms of a tolerance associated with the manufacturing system's *production rate,* r. Let the tolerance be given by $T = \{r/r \geq 7.5\}$, and assume that the anticipated *production rate,* r, obeys the normal probability law

$$f(r) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-(1/2)(r-\mu)/\sigma]^2} \tag{28}$$

with mean $\mu = 8$ and standard deviation $\sigma = 1.06$. Then the probability of success p is computed as follows:

$$p = 1 - \Phi\left(\frac{7.5-8}{1.06}\right) = 1 - \Phi(-0.47) = \Phi(0.47) = 0.6808 \tag{29}$$

where $\Phi(x) = (1/\sqrt{2\pi}) \int_0^x e^{-(1/2)\, t^2}\, dt$ is the standard normal probability distribution function. Thus the functional information content of the flexible manufacturing system is $F = \log_2(1/p) = 0.554$.

The foregoing approach is also used to define the design process functional complexity measure as the functional information content associated with the respective output solution. Thus, two design processes may be compared based on their outputs, such that the "best" design process is the one that yields an artifact in which its probability of successfully achieving the required functional requirements is maximized.

Finally, we show that the functional information content F as defined in (23) is consistent with the structural information content H as defined in (3). Indeed, assuming the operands and operators are independently (and sequentially) chosen, it was shown in Section IV that the probability that a particular design form (a "solution") may be found (a "functional requirement") in a certain design stage is: $p = (1/\eta)^L$. Thus, the functional information content associated with the particular design form is given by $\log_2 = 1/[(1/\eta)^L] = L \log_2 \eta$, in accordance with the structural information content provided in (3). The consistency between the structural and function complexity measures suggests that a measure based on the logarithm of the probability of success may be universal.

XIII. Summary

Starting from the evolutionary model of the design process proposed in [15] and [16], we gave two definitions of design complexity (structural complexity versus functional complexity), each leading to two types of value measures.

The proposed measures enable us to evaluate the complexity of a design artifact as well as the complexity of a design process. In the course of the design process, complexity measures may be utilized by designers for comparing alternative design forms and determining which path will be most efficient. Thus, during the design process, the measurable properties will be visible and can be continuously monitored. The proposed complexity measures also lead to the ability to rapidly estimate the approximate total assembly time of a product, and the manual assembly efficiency introduced by Boothroyd and Dewhurst in their DFA structured methodology [12]. The analogy between the design process and thermodynamics as shown in Section IV, serves to emphasize the limited but highly useful role of science in engineering. In other words, the measures presented here reveal how well a design form has been constructed, but they do not determine whether the design form should have been constructed in the first place. Instead, just as thermodynamics permits the engineer to calculate the maximum efficiency achievable with the optimal engine working between two specified temperatures; the design complexity measures enable the designer to calculate the "maximum efficiency" obtainable using the best possible design method working between two specified design stages.

References

[1] J. T. Carrol and T. F. Bellinger, "Designing reliability into rubber and plastic ac motor control equipment," *IEEE Trans. Ind. Gener. Applicat.,* vol. IGA-5, no. 4, pp. 455–464, 1969.

[2] R. L. Crouse, "Graphic trees help study of reliability versus cost," *Prod. Eng.,* vol. 38, pp. 48–49, 1967.

[3] J. Mihalski, "Design to cost—versus design to customer requirements—versus design for safe operation: Is there a conflict?," *Proc. ASME,* vol. 75-SAF-2, 1975.

[4] N. P. Suh, *The Principles of Design.* New York: Oxford Univ. Press, 1990.

[5] S. Pugh, "Load lines: An approach to detail design," *Prod. Eng.,* vol. 56, pp. 15–18, 1977.

[6] M. A. M. Mahmoud and S. Pugh, "The costing of turned components at the design stage," in *Proc. Information Designers Conf.,* Southampton, U.K., 1979, pp. 37–42.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.,* vol. 27, pp. 379–423, 1948.

[8] N. Wiener, *Cybernetics.* Cambridge, MA: MIT Press, 1948.

[9] J. M. Stroud, "The fine structure of psychological time," *Ann. New York Acad. Sci.,* pp. 623–631, 1966.

[10] H. A. Simon, and E. A. Feigenbaum, "A theory of the serial position effect," in *Models of Thought,* H. A. Simon, Ed. New Haven, CT, Yale Univ. Press, 1982, vol. 1.

[11] D. Ullman, T. G. Dietterich, and L. A. Stauffer, "A model of the mechanical design process based on empirical data," *AI EDAM,* vol. 2, pp. 33–52, 1988.

[12] G. Boothroyd and P. Dewhurst, *Product Design for Assembly.* Wakefield, RI: Boothroyd and Dewhurst, 1987.

[13] J. G. Bralla, *Handbook of Product Design for Manufacturing.* New York: McGraw-Hill, 1986.

[14] F. P. Ostwald, *Cost Estimation for Engineering and Management.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[15] O. Maimon and D. Braha, "On the complexity of the design synthesis problem," *IEEE Trans. Syst., Man, Cybern. A,* vol. 26, pp. 142–151, Jan. 1996.

[16] D. Braha and O. Maimon, "The design process: Properties, paradigms, and structure," *IEEE Trans. Syst., Man, Cybern. A*, vol. 27, pp. 146–166, Mar. 1997.
[17] ——, *A Mathematical Theory of Design.* Boston, MA: Kluwer, 1998.
[18] I. Ben-Gal, D. Braha, and O. Maimon, "A probabilistic sequential methodology for system redesign," in *9th Informs Applied Probability Conf.*, Boston, MA, 1997.
[19] T. J. McCabe, "A complexity measure," *Softw. Eng.*, vol. SE-2, no. 4, pp. 308–320, 1976.
[20] J. L. McTap, "The complexity of an individual program," in *Proc. 1980 NCC.* Arlington, VA: AFIPS Press, 1980, pp. 767–771.
[21] N. Chapin, "A measure of software complexity," in *Proc. 1979 NCC.* Arlington, VA: AFIPS Press, 1979, pp. 995–1002.
[22] M. H. Halstead, *Elements of Software Science.* New York: Elsevier/North-Holland, 1977.
[23] P. Barkan and C. M. Hinckley, "The benefits and limitations of structured design methodologies," *Manuf. Rev.*, vol. 6, pp. 211–220, Sept. 1993.

Learning While Solving Problems in Best First Search

Sudeshna Sarkar, P. P. Chakrabarti, and Sujoy Ghose

***Abstract*—We investigate the role of learning in search-based systems for solving optimization problems. Many AI problem solving systems solve problems repeatedly from the same domain. If the problems come from the same distribution in the learning phase and the problem solving phase, the problem solver can acquire information while solving problems, which can be used to solve subsequent problems faster. We use a learning model, where the values of a set of features can be used to induce a clustering of the problem state space. The feasible set of h* values corresponding to each cluster is called h*set. If we relax the optimality guarantee, and tolerate a risk factor, the distribution of h*set can be used to expedite search and produce results within a given risk of suboptimality. The off-line learning method consists of solving a batch of problems by using A* to learn the distribution of the h*set in the learning phase. This distribution can be used to solve the rest of the problems effectively. We show how the knowledge acquisition phase can be integrated with the problem solving phase. We present a continuous on-line learning scheme that uses an "anytime" algorithm to learn continuously while solving problems. The system starts with initial assumed distributions of h*set which are used to solve the initial problems. The results are used to update the distributions continuously and with time the distributions converge.**

***Index Terms*—Anytime algorithm, best first search, continuous learning, heuristic features, learning, problem solving.**

I. Introduction

An intelligent problem solving system can improve its performance by learning from past experience in solving problems. Our interest is in search-based problem solving systems that are required to find optimum/suboptimum solutions to optimization problems. A* [1] is a best first search that uses a heuristic function $h(n)$ at a node as an estimate of the optimum solution cost of the node n, denoted by $h^*(n)$. The algorithm works by expanding the most promising node next. The node selected next for expansion is the node with lowest f value, where the f value at a node n is calculated as $f(n, P) = g(n, P) + h(n)$. Here $g(n, P)$ is the cost of the current path P from the start node to the node n. If $h(n)$ is always an underestimate of $h^*(n)$, then the first solution obtained by running A* can be proved to be an optimum solution, and the search is said to be *admissible*. Learning a more accurate heuristic underestimate of a node helps to speed up best first search.

In this paper, we assume that prior information is available in the form of a set of features of the problem domain, where each feature is a heuristic function $h_i(n)$. Our framework of learning includes a scheme of clustering the state space, the acquisition of information corresponding to each class, and finally using a suitable algorithm to use the information effectively. In Section II, we briefly review previous work on learning in search. In Section III, we present a model of the learning system. In Section IV, we show how to use the distribution of optimum solution costs as the information for nodes in a class. In Section V, we present a scheme C_δ^* for integrating the knowledge acquisition with the problem solving phase. Experiments have been conducted in the domain of 8-puzzle and the experimental results corroborate our claims.

Manuscript received February 13, 1994; revised April 26, 1995; May 2, 1996; and October 3, 1997.

S. Sarkar is with the Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati, India 781 001 (e-mail: sudeshna@iitg.ernet.in).

P. P. Chakrabarti and S. Ghose are with the Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur, India 721 302 (e-mail: ppchak@cse.iitkgp.ernet.in; sujoy@cse.iitkgp.ernet.in).

Publisher Item Identifier S 1083-4427(98)02815-X.

II. Previous Work in Search Learning

The complexity of heuristic search algorithms depends on the accuracy of the heuristic evaluation function. Unfortunately effective underestimating heuristics are often hard to come by, or expensive to compute. The learning of heuristics can take place in two levels: 1) obtaining low-level features which can also be generated mechanically by abstraction of the problem; and 2) appropriate combination of available low-level features to give the heuristic information of the node. Heuristics can be discovered by consulting simplified or relaxed models of the problem domain [2]–[7]. Learning has also been used by search systems to find out an appropriate combination of low-level features to be used as heuristic estimate [8]–[15]. Given multiple features of the problem domain in the form of a *feature set* or *feature vector* $\vec{k} = (h_1, h_2, \cdots, h_k)$, best first search algorithms traditionally compute the values of the corresponding heuristic functions $[h_1(n), h_2(n), \cdots, h_k(n)]$ and combine the values in a certain way to yield a single value, h, that is used as the estimate (usually an underestimate) of $h^*(n)$ at node n. Given a set of underestimating features, the value of $\max[h_1(n), h_2(n), \cdots, h_m(n)]$ can be used as the estimate of a node in A*. But can we do better? Given multiple heuristic functions or features of a problem domain, the appropriate combination of the feature values that will yield a more informed estimate of a node; and at the same time yield an admissible search procedure, is an interesting problem. Some work in this regard has been done by Samuel [8], Lee and Mahajan [10], Christensen and Korf [11], and Bramanti-Gregor and Davis [12]–[14] who propose ways of combining the feature values. Most existing work in learning for search requires learning an appropriate algebraic formula of a given set of features, that can be used as an effective estimate. The method of de-biasing an inadmissible heuristics has been addressed by Pearl [4] and Chenoweth and Davis [16]. Bramanti-Gregor and Davis [12] have proposed a method for learning a good admissible heuristics given an admissible or inadmissible heuristic feature. Their