Systematic generation of constituent models of particle families

Raúl E. Valdés-Pérez [1] and Jan M. Żytkow [2]
[1] *Computer Science Department and Center for Light Microscope Imaging and Biotechnology, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213*
[2] *Computer Science Department, Wichita State University, Wichita, Kansas 67260-0083 and Institute of Computer Science, Polish Academy of Sciences, ulica Ordona 21, Warsaw, Poland*
(Received 7 February 1996)

A recurrent task in elementary particle physics involves building constituent models for particle families that can account for the particles' quantum numbers. We describe a systematic computerized approach to this task based on artificial intelligence principles, and discuss the output of two implemented programs that find constituent models of some standard families using additive quantum numbers. We also illustrate the capability to search for alternative quark models for single exotic particles in terms of the standard quarks. Although the current programs make use of very limited constraints on models, they serve as a foundation upon which more elaborate model-building programs can be designed. [S1063-651X(96)11707-4]

PACS number(s): 07.05.Mh, 12.39.−x, 12.60.−i, 12.90.+b

I. INTRODUCTION

The task of devising a constituent model for a family of particles recurs in elementary particle physics. The most familiar examples are quark structures for particle families such as baryons and mesons.

The constituent model explains existing particles by postulating their underlying constituents. The model also explains particle properties by summation or some other combinational rule applied to the property values of the constituents. The constituents should be as few as necessary and certainly much fewer than the members of the original family to be explained.

This article proposes an inferential mechanism for the development (discovery) of constituent models. We describe computer programs that can find simple constituent models and we apply the programs to some familiar particle families. Given a group of particles G described by their quantum numbers, the programs can postulate hidden constituents, their properties, and their combinations sufficient to explain G. The hidden constituents can be postulated from scratch, but some of them can also be given in the input. In the latter case, the programs make use of given constituents. If a simple model cannot be found, the programs consider more complex models by proposing additional constituents.

This paper will illustrate all these capabilities. We will show the inference of the standard quark model as well as a number of alternatives found by our programs: models for baryons and mesons, and models of quarks and leptons in terms of rishons. We will illustrate the incremental approach to model building by feeding a program a sequence of particle families, one at a time, and using the quarks introduced for the previous families to construct models for the new ones. We will also show a construction of the simplest model of a specified particle in terms of the standard quarks. For example, we will use two excited particle states (hexaquark dibaryon and diquonium) to find their simplest models in terms of known quarks.

A model may propose additional particles in addition to those provided in the input. We will illustrate this capability with the prediction of Ω^- in the baryon resonance family.

These programs, together with recent work on finding phenomenological rules of conservation and new quantum properties [1–3], illustrate the potential of heuristic search principles and knowledge representation in artificial intelligence [4], applied to high energy physics. The scope of computing in high energy physics has been traditionally concentrated on the early stages of interpreting accelerator data, and on numerical and symbolic mathematics (e.g., [5,6]). The work we present helps to extend computer systems into the realm of model building and discovery. The present paper follows up on a preliminary report to an artificial intelligence audience [7]. We use additional examples to show various advantages of the method in physics.

II. TASK FORMULATION

The vast majority of discovery problems cannot be solved by algorithms that directly lead to the goal. Discoverers explore possibilities which, from the perspective of eventually accepted solutions, can be called dead ends because they do not become parts of those solutions. A search typical of discovery programs is a process of gradual construction and evaluation of alternative hypotheses and models, which are generated by exploring alternative decisions at various choice points. In line with this discovery process, the states that are reached are tested against the available data, background theory, and constraints; the states that fail the tests are abandoned. Some effort is usually made to apply tests as soon as they become relevant, although some tests will be applicable only to fully specified hypotheses. By the early application of tests, the unpromising partial models can be discarded, so that the overall search is substantially reduced, which may be crucial in practice. Those partial models which pass the evaluation are further elaborated and tested so that eventually some complete solutions may be generated.

This process of search within a problem space (also called a search space or state space) has been developed in artificial intelligence into a conceptual tool to enable a unified treatment of problem solving and discovery processes [4,8].

TABLE I. Strange baryon octet (omitting one particle): Two models generated by GELL-MANN.

	Input family			Output quarks	
Particle	Charge	I_3	Strangeness	Model 1	Model 2
n	0	$-1/2$	0	*udd*	*bcc*
p	$+1$	$+1/2$	0	*uud*	*acc*
Σ^-	-1	-1	-1	*dds*	*bbc*
Σ^0	0	0	-1	*uds*	*abc*
Σ^+	$+1$	$+1$	-1	*uus*	*aac*
Ξ^-	-1	$-1/2$	-2	*dss*	*abb*
Ξ^0	0	1/2	-2	*uss*	*aab*

Output models

	Model 1				Model 2		
Quark	Charge	I_3	Strangeness	Quark	Charge	I_3	Strangeness
$a(u)$	$+2/3$	$+1/2$	0	a	$+1/3$	$+1/2$	$-2/3$
$b(d)$	$-1/3$	$-1/2$	0	b	$-2/3$	$-1/2$	$-2/3$
$c(s)$	$-1/3$	0	-1	c	$+1/3$	0	$+1/3$

At this point, we recommend that the reader look at Table I for a concrete example that will motivate the next sections on task formulation and program design. The input particle family (strange baryon octet—upper left of the table) will be the input to our computer programs, which will output simple constituents (quarks—upper right), and constituent models of the particles in terms of these quarks (bottom of table).

A. Constituent models

By a constituent model of a particle family we will understand specification of the following:

1. A set $T=\{t_1,\ldots,t_N\}$ of postulated constituents, representing N different constituent types.

2. A set C of admissible macro-objects, each defined as a bag (multiset) of constituents from T. $C=\{c|c=[n_1\times t_1,\ldots,n_N\times t_N]$ and $\varphi(n_1,\ldots,n_N)\}$, where the t_i are drawn from T, the n_i are non-negative integers (each n_i states how many copies of t_i are in c), and $\varphi(c)$ is a constraint on admissible bags. For instance, the constraint $\Sigma_{i=1}^{N} n_i=M$ states that each bag contains the same number M of postulated constituents.

3. A set $P=\{P_1,\ldots,P_K\}$ of properties for constituents in T and macro-objects in C.

4. (Optionally) A set V_i of admissible values that constituents in T may have, for each property P_i in P.

5. Specific property values for each postulated constituent and property, $P_i:T\rightarrow V_i, i=1,\ldots,K$. That is, each postulated constituent in T has an admissible property value. Each pair of constituents in T should differ by at least one property value.

6. For each property P_i in P, the relation between the value of P_i for each macro-object c in C and the values of P_i for the constituents of c from T. In this paper, we use the additivity principle: For each object c in C, and all constituents $c_1,\ldots,c_M$ of c from T, $P_i(c)=\Sigma_{j=1}^{M} P_i(c_j)$.

7. A mapping from the input particle family onto the set C of admissible macrostructures. This mapping assigns to each particle a bag of constituents from T.

The above specification can lead to different problems of model construction, depending on what aspects of a constituent model are given, and what other aspects are to be inferred. For example, our computer programs, to be described below, can at the same time postulate a set of quark constituents T (item 1 above), the number of quarks that constitute a given particle (the constraint in item 2), property values for each quark (item 5), and the mapping between particles and bags of quarks (item 7). These elements of the model are derived from the set P of properties (item 3), from a family of particles and their property values, from the additivity principle (item 6), and (optionally) from the admissible set of property values V_i for each property P_i (item 4).

One could devise further model-building tasks by focusing on other properties of the input particles, on alternative schemes for combining property values (e.g., vector additivity), and on other constraints on admissible bags.

B. Search space

The space of constituent models can be represented by the matrix equation in Fig. 1, in which the contents of matrices $\mathcal{S}$ and $\mathcal{P}$ are filled in by a combinatorial search, while the matrix equality enforces the additivity principle. This representation has proved useful to describe the search space and solution constraints of various model-building tasks in science [9].

The $\mathcal{Z}$ matrix contains the initial data, which consist of the property values of a particle family. The properties (e.g.,

$\mathcal{S}$	a b c →
P	
Q	(Fill in)
⋮	
W	

$\times$

$\mathcal{P}$	$P_1 \ldots P_K$
a	
b	(Fill in)
c	
↓	

$=$

$\mathcal{Z}$	$P_1 \ldots P_K$
P	
Q	(Given)
⋮	
W	

FIG. 1. The search space for constituent models of particles.

charge, isospin, etc., listed generically as $P_1, \ldots, P_K$) are represented by the columns of $\mathcal{Z}$, and the rows describe particles, depicted generically as $P, Q, \ldots, W$.

The $\mathcal{S}$ matrix describes how the known particles are made up of constituents, the latter represented generically as *a*, *b*, *c*, and so on. For example, an entry of 2 in the row *P* and column *b* means that the particle *P* contains two *b* constituents. Every entry in $\mathcal{S}$ is a non-negative integer, and further constraints can be imposed on $\mathcal{S}$. The programs described below postulate that every particle in a given family contains the same number of constituents, i.e., the sum of row entries in $\mathcal{S}$ is uniform for every row.

The $\mathcal{P}$ matrix expresses the property values for the constituents in a similar manner as the matrix $\mathcal{Z}$ does for the particles.

In this paper, the additivity principle is an invariable constraint on constituent models: the arithmetic product of the matrices $\mathcal{S}$ and $\mathcal{P}$ equals the matrix $\mathcal{Z}$.

In all cases the $\mathcal{Z}$ entries will be input to our programs and the $\mathcal{S}$ entries are to be filled in subject to constraints, whereas on different tasks various parts of the $\mathcal{P}$ matrix are given while the remainder are to be filled in. Since the property values in $\mathcal{Z}$ are rational numbers and the numbers in $\mathcal{S}$ are non-negative integers, the entries in $\mathcal{P}$ are also rational. Furthermore, for fixed matrix dimensions and filled-in $\mathcal{S}$ and $\mathcal{Z}$ matrices, the equation $\mathcal{SP}=\mathcal{Z}$ has in practice a unique least-squares solution, because most rows of $\mathcal{S}$ will be distinct (few or no rows will be linearly dependent), and the number of rows (particles) in $\mathcal{S}$ will much exceed the number of columns (constituents). Hence, the problem is overconstrained, thus leading to a unique solution as implied by an elementary theorem of matrix algebra. (Of course, the least-squares solution need not imply an *exact* solution to the equation $\mathcal{SP}=\mathcal{Z}$.)

The number of rows in $\mathcal{P}$ is not known whenever the constituents need to be hypothesized from scratch; the arrow notation ($\rightarrow$ and $\downarrow$) signals the dimensions along which the matrices grow, when new constituents are postulated. This inferential task is combinatorially the hardest because the space of open possibilities (the entire $\mathcal{S}$ and $\mathcal{P}$ matrices) is, by far, the largest.

A second task involves searching for alternative constituent models made of known constituents for particles which are already known to have at least one model. In this case, the matrix $\mathcal{P}$ is already filled in with the properties of known constituents, and the task only involves searching the $\mathcal{S}$ matrix, which makes this combinatorially the simplest task.

A third inferential task involves finding a constituent model for a particle family, while making use of a given set of constituents which *might not be sufficient.* In this case, new constituents may need to be postulated as well. This *incremental* search mode arises typically when quarks found for a previous particle family are the starting point for explaining a new family. The $\mathcal{P}$ matrix is partially filled in at the start, but new rows can be added during the search. This third task is of intermediate combinatorial difficulty.

These three different inferential tasks will be illustrated below on concrete examples taken from physics practice.

C. The role of simplicity

The search for constituent models ordinarily involves finding the simplest model or the set of all simplest models. One measure of simplicity is the number of constituents, which normally starts at the smallest plausible value and is incremented as simpler models are rejected. As the number of constituents increases, both of the matrices $\mathcal{S}$ and $\mathcal{P}$ in Fig. 1 are successively enlarged along one of their dimensions: the number of postulated constituents.

A second simplicity parameter is the number of constituents per particle, e.g., the number of quarks that make up an elementary particle, which is assumed here to be equal for all the particles in a family. This parameter is expressed as a constraint on the rows of the matrix $\mathcal{S}$: the sum of row entries must equal an integer greater than 1. This second simplicity parameter is also subject to a search process, which starts from the value 2 and increments the parameter by 1 as simpler models fail to turn up solutions.

The quantum numbers of elementary particles are often limited to a small set of fractions and integers. We might similarly limit the admissible property values for constituents. If the search for models is constrained to a given set of admissible values for each property in *P*, then this constraint might be considered another simplicity criterion, especially if the set of admissible values is expanded dynamically whenever no solutions are found using a smaller set. The constraint on admissible values reduces the number of possible $\mathcal{P}$ matrices, but this number is still potentially very large. For three constituents, three properties, and seven possible values for each property, the number of possible matrices is $7^{3\times 3}\approx 4\times 10^7$.

III. COMPUTER PROGRAMS

Next we will describe two computer programs that search the space of constituent models [18]. Each model is constructed gradually. The construction steps correspond to categories 1–7 in our hidden structure definition in Sec. II A. Each element of the definition is represented by operators that build the corresponding part of the model: a number of constituents are postulated, as are their properties, the combinations of constituents into particle structures, and so forth. Alternative models are constructed by following alternative paths that are enabled at the various choice points.

A. Search requirements

Since model construction is a gradual process, the search for hidden structure is actually conducted in the space of partial models. This is very important for search efficiency (consider an alternative approach that postulates complete constituent models out of whole cloth: much material and time will be wasted before a fit is found). Partial models are evaluated as early as possible by the available data and constraints, to prevent the further (exponential) elaboration of failed models.

Another important requirement is that the search not overlook any solutions. Each possibility must be tried until it is clear that it cannot be expanded into an acceptable model. On the other hand, ideally no possibility should be tried more than once, and isomorphic solutions should be excluded, that is, solutions that can be mutually mapped by renaming the constituent types in *T*. If possible, the model generator should be isomorph-free, that is, only one model should be tried in each isomorphism class.

The search carried out by our computer programs will start with the simplest models and move to more complex models only after an exhaustive search in the simpler classes fails to find any solutions. This strategy is called breadth-first search. The two main simplicity parameters are the number N of constituent types (e.g., quarks) and the number M of constituents in each structure (quarks per particle).

B. From structure definition to search strategy

In response to the first item in our model definition, a program proposes the number N of constituents to consider. It starts from $N=2$, and increases the number by 1 if no solution has been found for a given N. In terms of the matrices in Fig. 1, each choice of N corresponds to a particular column dimension in matrix $\mathcal{S}$ and row dimension in matrix $\mathcal{P}$. At $N=k$, where k is the number of particles in the input family, many models always exist, including a trivial model, which is the input family itself. Hence, the search stops after N reaches $k-1$. If there is ''pattern'' in the input family, then one expects to find a model that has much fewer constituents than the trivial model.

The second item calls for creation of all admissible structures. We use the constraint that requires the same number M of constituents for every particle in the input family. The initial value of M is 2, because $M=1$ could only lead to a trivial model. As is the case with N, M is increased by 1 if no model is found for a given M. The constraint that M is uniform over the entire input family is admittedly *ad hoc*. Looser constraints on structure could be used, or alternatively, a space of constraints, but that could lead to a much longer search.

For a given number N of postulated constituents and M constituents per particle, a formula from elementary combinatorics indicates that there are

$$|C|=\binom{N+M-1}{M}$$

different combinations (bags) made up of constituents (duplicates are allowed in a bag). These structural combinations are the only admissible candidates for the rows in the $\mathcal{S}$ matrix. For example, if $N=4$, so there are four constituents a, b, c, and d, and if $M=2$, then there are 10 possible combinations:

$$(aa\ \ ab\ \ bb\ \ ac\ \ bc\ \ cc\ \ ad\ \ bd\ \ cd\ \ dd).$$

If the number k of particles in the input family is less than

$$\binom{N+M-1}{M},$$

(here, 10), then of course some of these combinations will be left unused in $\mathcal{S}$.

We can impose further constraints on N and M, given the size of the input particle family. First, $|C|$ must be no less than the number of particles in the family, or else some particles could not be accounted for. Second, if we want to disallow too many potential combinations in addition to those which explain the k input particles, then $|C|<3k$ seems a reasonable constraint. In conclusion, for a given N, the number M has a tight upper and lower limit, namely, those integers that satisfy

$$k\leqslant\binom{N+M-1}{M}<3k.$$

Since $N<k$, only a small finite set of N,M pairs will pass those tests.

For a fixed N,M simplicity class, our programs will search for acceptable models. If none is found, then M is incremented preferentially, not N, since it seems more important to minimize the number of constituents than the number of constituents per particle. (Of course, one could run the program under different priority schemes and compare the results.) If

$$\binom{N+M-1}{M}$$

reaches $3k$ (or another such threshold), then the search should increment N and reset M to its minimal value of 2. After either M or N has been incremented, the search for acceptable models proceeds in the newer, more complex class.

The constraint of uniform M on admissible structures is *ad hoc* and many alternative constraints are possible. In the future we plan to expand the search into the space of constraints.

The third item in our model definition, the list of the constituent properties, is taken directly from the input matrix $\mathcal{Z}$ and forms the column dimension of the matrix $\mathcal{P}$. That is, the constituent properties exactly parallel the particle family properties. The task of postulating *new properties*, in contrast to new property *values*, is not addressed here (but see [1] and [2,3,10]). The $\mathcal{Z}$ matrix contains only the additive properties of elementary particles, due to the linear equation. Other nonadditive properties could be handled within an expanded framework.

The fourth, fifth, and sixth items deal with assigning quantum numbers (property values) to constituents and testing the assignment by additivity. These steps are handled differently by the two programs to be described below.

The seventh item involves search in the space of possible mappings between particles and bags of constituents. Potentially, for k particles mapped to s constituent combinations,

$$s=\binom{N+M-1}{M},$$

the number of mappings is $s\times s-1\times\cdots\times s-k+1$. To illustrate, this number is 3.6×10^{6} for the baryon decuplet modeled by three quarks in groups of 3. To reduce the combinatorial overhead, our programs construct the mappings incrementally, in parallel with assigning property values to constituents.

There is a potential for vastly redundant search if isomorphic solutions are not excluded. Suppose that the current simplicity class involves four constituents in groups of two ($N=4$, $M=2$); let us consider the first assignment of con-

stituents to a particle, i.e., the first row of the $\mathcal{S}$ matrix. One possible assignment is the pair *aa*. Another assignment *bb* is equivalent to *aa*, since both *a* and *b* are mere interchangeable labels at this early stage; they become distinct after *a* or *b* is assigned a property value, or after *a* or *b* is assigned to the components of a particle without yet committing to specific property values. However, the third option *ab* is *not* equivalent to *aa*, since the former does not entail double occurrence of a constituent. A good structure generator should attend to such subtleties and not generate redundant possibilities. For $N \geqslant 2, M = 2$, the only nonredundant mappings at the first assignment are *aa* and *ab*.

Our programs generate isomorph-free constituent models according to three principles that govern interchangeable labels: (1) constituents are listed in alphabetic order; (2) no constituent is skipped; (3) a next constituent (in alphabetic order) can be listed no more times than the previous constituent. For example, when $N = 5$ and $M = 4$, only the combinations *aaaa*, *aaab*, *aabb*, *aabc*, and *abcd* are considered for the first particle.

Finally, both our programs report all satisfactory simplest models by carrying out an exhaustive search; they do not stop at the first model found within a simplicity class. Alternatively, the programs can start with user-specified parameters N and M.

C. GELL-MANN: Divide-and-conquer and trial-and-error

GELL-MANN, the first program we describe, uses a divide-and-conquer approach to handle the large combinatorial space spanned by the unknown entries in matrices $\mathcal{S}$ and $\mathcal{P}$. The search is decomposed into parts that can be solved separately, then the partial solutions are joined into an overall constituent model. Further, GELL-MANN uses trial and error to find the right property values in $\mathcal{P}$.

The third item in our model, the list of admissible values for each property, is determined by the quantum numbers of input particles in $\mathcal{Z}$ [7]. For each property P_i, let v_i be the highest absolute value for the P_i column in $\mathcal{Z}$, and let D_i be the set of all denominators among fractional values for the same column. The list V_i of admissible values then consists of all integers and proper fractions based on denominators in D_i, between v_i and $-v_i$. For instance, because the maximum value of strangeness is 2 for the hadron octet and all values of strangeness are integers (see Table I), the program considers the list of values $V_{strange} = (2,1,0,-1,-2)$. Because the maximum value of isospin for the hadron octet is 1, and 2 is the denominator in some fractional values, $V_{isospin} = (1,1/2,0,-1/2,-1)$. When the program cannot find a solution for those values, it considers the fractions based on 1/M. Thus, if $M = 3$, and GELL-MANN cannot find a model for the charge values $1,0,-1$, the program eventually considers the values $1,2/3,1/3,0,-1/3,-2/3,-1$.

GELL-MANN's choice of admissible values has been historically motivated. For instance, initially the integer values were considered admissible for charge and only reluctantly have the fractional values of 1/3 and 2/3 been accepted. The values admissible for strangeness are still integers, while our programs find also models with values of 1/3 and 2/3.

GELL-MANN decomposes the search for the property values of constituents by first considering each property separately and finding all partial solutions for that property alone. Clearly, not every partial solution will become part of an overall solution, so GELL-MANN then tries to unify the partial solutions by successive merges.

D. YUVAL: Search and equation solving

After GELL-MANN was implemented and applied [7], a second program, YUVAL, was written by the first author to apply the concept of model building as search in matrix spaces [9]. This concept was used in Sec. II B to characterize the search space for constituent models. Here we discuss only the main differences between the two programs.

The newer program does not attempt a divide-and-conquer approach based on lists of possible property values inferred from the $\mathcal{Z}$ matrix, although both programs search the $\mathcal{S}$ matrix (recall that the rows of $\mathcal{S}$ correspond to particles and the columns to constituents). The search proceeds by filling in a row at a time (i.e., building up a partial model $\mathcal{S}'$). Whereas GELL-MANN tests combinations of admissible values, YUVAL tests whether the matrix equation $\mathcal{S}' \times \mathcal{P} = \mathcal{Z}'$, in which $\mathcal{P}$ is the only unknown, is soluble. If the equation is soluble, then YUVAL continues its search by considering a constituent model for the next particle (i.e., row in $\mathcal{S}$) remaining to be considered, or stores the solution if all particles were already considered. If the equation is insoluble, then the program considers alternative constituent models for the last row or rows added to $\mathcal{S}'$.

The process of solving the matrix equation yields the desired property values of constituents, that is, the $\mathcal{P}$ matrix. Of course, any calculated $\mathcal{P}$ is not a final solution unless all the particles in the family were included in $\mathcal{S}'$.

Whenever the rank of the filled-in rows of $\mathcal{S}'$ equals the number of columns (i.e., postulated constituents), and if $\mathcal{S}' \times \mathcal{P} = \mathcal{Z}'$ is soluble, then the constituent property values in $\mathcal{P}$ are unique for this $\mathcal{S}'$. In such cases, the values in $\mathcal{P}$ can be used to quickly test whether the available quark combinations for the remaining rows (particles) in $\mathcal{S}$ are acceptable. The effect is a very large reduction in the search time on larger particle families, since typically only N rows will need to be filled in to reach a rank of N.

YUVAL carries out an exhaustive search of an (N,M) simplicity class in the sense that, barring program bugs, it misses no constituent model, even if it involves exotic fractional values; this feature is its principal advantage. However, empirical comparisons between YUVAL and its predecessor GELL-MAN reveal that the latter is significantly faster due to its divide-and-conquer strategy and use of small lists of values. GELL-MANN may be able to tackle some problems that are beyond the combinatorial ability of YUVAL, although the latter program may find models that are outside the scope of the former.

E. Incremental model construction

Suppose an acceptable constituent model has been found for one particle family, and a new family of particles presents itself. One can then use the current constituents to find a model for the new family, and this echoes the historical practice in physics.

Our programs can operate in this mode, in which the $\mathcal{P}$ matrix is also an input, by only varying M during their

search within $\mathcal{S}$ [7]. If a model cannot be found, then the programs postulate novel constituents, exactly as described earlier. In this latter case, the final $\mathcal{P}$ matrix is only partially filled in at the start.

The two families could be merged at the start so that a joint model is generated from scratch. This might be undesirable, however, because nonincremental search in both programs operates under the constraint that the number of constituents per particle be uniform, whereas one might prefer to allow different M values for the different particle families.

F. Limitations

In the current programs, a particle's properties are determined only by simple summation of the properties of its constituents. The programs do not take into account quantum numbers that are added as vectors, for instance, angular momentum and isospin.

The current framework can be extended to express further constraints on constituent models or property values. For example, suppose that constituents must exist in complementary pairs, such that constituent quantum numbers are of equal magnitude but opposite sign to those of its anticonstituent. Then the $\mathcal{P}$ matrix will have an even number of rows, and the complementarity can be expressed as a linear constraint on $\mathcal{P}$. Neither of the above capabilities has been implemented within our computer programs, although incorporating them does not seem problematic.

Lastly, the programs confront a combinatorial explosion with increases in the numbers of particles, constituents, and constituents per particle. This explosion should not be seen as defects of program design, but as inherent to exhaustive search in combinatorial spaces. Further constraints on models could reduce the combinatorics, but they may also eliminate interesting solutions from the scope of the search.

IV. APPLICATION TO PARTICLE FAMILIES

GELL-MANN and YUVAL can be viewed as convenient tools to propose underlying constituent models for various groups of particles, including particle families. We will now consider various applications of the programs, and highlight especially the exotic results that differ from the standard model. The run times for the two programs, both of which are written in Common Lisp, vary from seconds to overnight on a workstation.

A. Strange baryon octet

Our first family is commonly referred to as the strange baryon octet or hadron octet and consists of seven unique particles. An eighth particle has the same quantum numbers as one of the seven particles and is not considered here. The input [11] and the results are summarized in Table I.

The first hypothesis for N and M that leads to at least seven quark combinations is three quarks in groups of three ($N=M=3$). GELL-MANN arrived at the two solutions shown in Table I. The first solution corresponds exactly to the quark model developed by Gell-Mann. The second model uses strangeness values that are multiples of 1/3, like the charge values in the first model.

TABLE II. Constituent models for all eight baryons.

Quark	Model 1			Model 2			Model 3		
	Q	I_3	S	Q	I_3	S	Q	I_3	S
a	1/2	1/2	−1/2	1/2	1/4	0	1	3/4	0
b	1/2	0	1/2	−1/2	−3/4	0	0	−1/4	0
c	−1/2	−1/2	−1/2	−1/2	−1/4	−1	−1	−3/4	−1
d	−1/2	0	−3/2	1/2	3/4	−1	0	1/4	−1

In each model, three possible quark combinations are in surplus, i.e., are not assigned to any of the seven particles. Each of these combinations consists of three identical quarks, for instance, *aaa*. These surplus combinations can be viewed as predictions of the model. The three extra particles in GELL-MANN's first solution have properties equal to baryon resonances Δ^-, Δ^{++}, and Ω^-. We could not find the particles predicted by GELL-MANN's second solution.

We also used all eight baryons, including Λ^0, which has the same properties as Σ^0 (Λ^0 and Σ^0 are thus required to possess distinct structural models). GELL-MANN could not find any solutions in the category ($N=M=3$), but it produced three quark solutions for $N=4$ and $M=2$, as shown in Table II. The first model predicts two extra particles. The properties of one are identical with Ω^-, whereas the other has a charge of 1, an I_3 value of 0, and a strangeness of 1.

B. Meson octet

According to the standard model, each meson (pion) is made up of one quark and one antiquark (Table III). For most properties, including charge q, strangeness S, and isospin I_3, the value of the antiquark is opposite in sign but equal in magnitude to the corresponding quark.

Surprisingly, for the meson family [12,13] GELL-MANN found a single solution for just four quarks in groups of two, presented as the first model in Table III. It can be interpreted as two quarks (a and b) and their two antiquarks ($\overline{a}$ and $\overline{b}$). None of these quarks is recognized in physics. This model predicts the particles bb ($q=1$, $I_3=0$, $S=2$) and $\overline{bb}$ ($q=-1$, $I_3=0$, $S=-2$).

YUVAL found two additional models of four quarks in groups of two (models 2 and 3 in Table III). These also consist of two quarks and their antiquarks, but the magnitudes of I_3 are 1/4 and 3/4. Both of these models also predict two extra particles (bb and $\overline{bb}$ in model 2, and aa and $\overline{aa}$ in model 3).

When we removed meson η (input identical to π^0), GELL-MANN discovered two models ($N=M=3$) analogous to the models discovered for seven baryons (add 1/3 to the strangeness value of each quark in Table I).

In the absence of additional constraints on quark models, it would be difficult for GELL-MANN or YUVAL to discover the standard six-quark model for mesons, for the following reason. When we forced GELL-MANN to search for five-quark models (instead of the simplest models), the program found hundreds of solutions with $M=2$, and it would find many more six-quark models. This result illustrates the underdetermination of models by observational data that occurs when additional hidden objects are permitted beyond the available minimum. Even if the standard six-quark model

TABLE III. Meson octet: Three models generated by GELL-MANN.

	Input family			Output models			
Particle	Charge	I_3	Strangeness	Standard model	Model 1	Model 2	Model 3
π^+	+1	+1	0	$u\bar{d}$	$a\bar{a}$	$a\bar{b}$	$a\bar{b}$
π^-	−1	−1	0	$\bar{u}d$	$\bar{a}a$	$\bar{a}b$	$\bar{a}b$
π^0	0	0	0	$\bar{u}u\ \bar{d}d\ \bar{s}s$	$a\bar{a}$	$a\bar{a}$	$a\bar{a}$
K^+	+1	+1/2	+1	$u\bar{s}$	ab	aa	$a\bar{b}$
K^0	0	−1/2	+1	$d\bar{s}$	$b\bar{a}$	$a\bar{b}$	$\bar{b}\bar{b}$
$\bar{K}^0$	0	+1/2	−1	$\bar{u}s$	$a\bar{b}$	$b\bar{a}$	bb
K^-	−1	−1/2	−1	$\bar{d}s$	$\bar{a}b$	$\bar{a}a$	$b\bar{a}$
η	0	0	0	$\bar{u}u\ \bar{d}d\ \bar{s}s$	$b\bar{b}$	$b\bar{b}$	$b\bar{b}$

Output quarks

	Model 1			Model 2			Model 3		
Quark	Charge	I_3	Strangeness	Charge	I_3	Strangeness	Charge	I_3	Strangeness
a	+1/2	+1/2	0	+1/2	+1/4	+1/2	+1	+3/4	+1/2
b	+1/2	0	1	+1/2	+3/4	−1/2	0	+1/4	−1/2
$\bar{a}$	−1/2	−1/2	0	−1/2	−1/4	−1/2	−1	−3/4	−1/2
$\bar{b}$	−1/2	0	−1	−1/2	−3/4	+1/2	0	−1/4	+1/2

could be generated, it would be one of many, so there would be no grounds to claim that it had been discovered. However, next we will show that the standard model for mesons can be discovered by incremental model building.

C. Incremental search: Baryons followed by mesons

As already stated, our programs can consider particle families incrementally, such that the constituents developed for one family become a starting point for a second family. When GELL-MANN is run on the baryon octet, it finds two models involving three quarks, which were presented in Table I. Then GELL-MANN is given the meson octet as the next input. The program tries to use the three quarks from model 1 (Table I) to explain the mesons, but fails. As a result, GELL-MANN postulates a new (fourth) quark, but still fails to find an acceptable model, so it introduces a fifth quark, and then a sixth quark. With these six quarks, three old and three new, GELL-MANN finds two models, including the standard model of three quarks and three antiquarks. (Actually, the number of models is larger because each of the mesons π^0 and η can be explained by three pairs of quark-antiquark made from the same quarks.) When GELL-MANN begins with model 2 (Table I), it again augments it with three new quarks, resulting in two models of six quarks in combinations of two.

YUVAL followed a slightly different path. It tried models for $N=5$, $M=3$, which GELL-MANN skipped because the number of candidate particles, that is, 35, exceeded three times the number of input particles. Starting with either model 1 or model 2, YUVAL found 42 extended models, many of them using different combinations of the same quarks to represent mesons π^0 and η.

If one imposes a further constraint, that the quarks come in complementary pairs, then six quarks are needed. YUVAL then finds a unique set of quark-antiquarks for each of the two original baryon models, which of course includes the standard model.

D. Baryon resonance family

For the decuplet of baryon resonances, both programs find only the standard model of three quarks in combinations of three. The second model (with fractional strangeness of 1/3 and 2/3) for the baryon octet, which was expandable to the meson octet, cannot by itself explain the full decuplet of baryon resonances.

E. Rishons

Harari [14] and Shupe [16] postulated a hidden layer of structure beneath the four quarks u, $\bar{u}$, d, $\bar{d}$ and four leptons e^-, e^+, $\ni_e$, $\bar{\ni}_e$. Harari called them rishons while Shupe named them quips. The only additive property used in both models is charge. Both models postulate two subquarks (of 0 and 1/3 charge), and their two antiparticles. Our programs found $N=3$, $M=3$ models and verified that no simpler structures exist for the quarks and leptons. It turns out that three distinct sets of three subquarks possessing the charge

TABLE IV. Subquark constituent model.

Quark	Charge	I_3	Strangeness	Charm	Model
d	−1/3	−1/2	0	0	*vwx*
u	2/3	1/2	0	0	*vvy*
s	−1/3	0	−1	0	*wzz*
c	2/3	0	0	1	*vvz*
$\bar{d}$	1/3	1/2	0	0	*vyz*
$\bar{u}$	−2/3	−1/2	0	0	*wxz*
$\bar{s}$	−1/3	0	1	0	*xyz*
$\bar{c}$	−2/3	0	0	−1	*wxy*
v	1/3	0	0	1/3	
w	−1/3	0	−1	−2/3	
x	−1/3	−1/2	1	1/3	
y	0	1/2	0	−2/3	
z	0	0	0	1/3	

TABLE V. Quark models of four diquonium states.

Particle	Baryon number	Charge	(Total) Isospin	Strangeness	Models
$D1$	0	1	3/2	−1	$us\overline{dd}$
$D2$	0	0	3/2	−1	$ds\overline{dd}$ or $us\overline{ud}$
$D3$	0	−1	3/2	−1	$ds\overline{ud}$ or $us\overline{uu}$
$D4$	0	−2	3/2	−1	$ds\overline{uu}$

values (−1/3, 0, 1/3), (−1/3, 0, 2/3), and (−2/3, 0, 1/3), can be arranged in 12 structural models. When one uses the property I_3 in addition to charge, the programs find the simplest models at $N=4$, $M=3$, all involving the unusual charge denominator of 9. A search at $N=4$, $M=4$ turned up 407 models. None of these models consisted of rishon-antirishon pairs.

F. Other subquark models

We ran YUVAL on the set of quarks shown in the first eight rows of Table IV. The program failed to find any constituent models for four subquark constituents, despite allowing models having as many as six constituents per quark. However, when five constituents were allowed, the program did find 40 different models having three constituents per quark (a smaller M did not turn up any models). From these 40 models, the last five rows of Table IV show one of the four models (v, w, x, y, z) that involve no denominator larger than 3. None of the 40 models involved strictly constituent-anticonstituent pairs.

G. Models of individual particles using standard quarks

As final examples, we applied YUVAL to the task of finding constituent models of individual particles (or various of their excited states), while drawing on the standard quarks, and not postulating further constituents. In the notation of Fig. 1, $\mathcal{P}$ and $\mathcal{Z}$ are given, and only $\mathcal{S}$ is to be filled in. As examples, we consider the cases of the exotic particles hexaquark dibaryon [15] and diquonium states [17].

The first example (the hexaquark dibaryon) has the properties of zero charge, a baryon number of 2, strangeness of −2, and has a proposed quark structure *udsuds* [15]. The program was given this particle as a singleton particle family (the $\mathcal{Z}$ matrix), together with the six quarks u, d, s and their antiquarks, with the standard quantum numbers of charge, baryon number, and strangeness. The program confirmed that the six-quark model *udsuds* was the simplest, in the sense of requiring the fewest number (zero) of new constituents, and the fewest number of constituents per particle. Moreover, the six-quark model was alone in its simplicity class of $M=6$.

Our second example consists of the four exotic particles (diquonium states) labeled $D1$–$D4$ in Table V. Drawing on the standard quarks, the program finds four different quark models which differ only in their assignment to two of the particles (last column of Table V). The two models each for $D2$ and $D3$ are recognized as mixed states for these two particles [19].

In both examples, YUVAL ruled out the possibility of alternative quark models, which seems difficult to guarantee without an exhaustive computerized search.

V. CONCLUSION

This paper has analyzed the task of postulating constituent models for particle families, and has described two separately developed computer programs capable of performing the task. We have used these programs to infer constituent models for the baryon and several other families, for quarks, and for individual exotic particles. In some cases, the programs detected several alternative models, in addition to the standard model, that might be of potential interest in the phenomenology of elementary particles.

The approach may also be useful to explore subquark models and models for exotic particles in terms of the standard quarks, as was done on the hexaquark dibaryon and diquonium states. In principle, the same approach works on ''particles'' at both higher and lower levels of aggregation than elementary particles and quarks, as long as additive properties are involved. For example, one could describe the large spectrum of atomic nuclei in terms of proton and neutron constituents. The same approach as used for a dibaryon would work on any single nucleus. However, the constraint on equal number of constituents might lead to strange results in the case of several joint nuclei.

From the physics perspective, probably the most significant result is that an exhaustive search in the space of quark models for the baryons followed by the mesons (Sec. IV C) reveals that the standard quark model stands out nearly uniquely as the simplest, when the constraint of complementary pairs is imposed. It would be difficult to arrive at such conclusions without the aid of the computer. Finally, we have laid a foundation for future discovery tools that can take account of a wider array of constraints and background theory than was addressed here.

ACKNOWLEDGMENTS

Many thanks to Paul Fischer for implementing the initial GELL-MANN system and for many experiments on the generation of quark models. Malcolm Perry contributed many helpful comments and suggestions. Ulrich Müller suggested the hexaquark dibaryon and diquonium examples, and commented on YUVAL's output. R.E.V.P. was supported in part by Grant No. IRI-9421656 from the (U.S.A.) National Science Foundation and by the Carnegie Mellon Center for Light Microscope Imaging and Biotechnology.

[1] S. Kocabas, Mach. Learn. **6**, 277 (1991).
[2] R. E. Valdes-Perez, Mach. Learn. **17**, 47 (1994).
[3] R. E. Valdes-Perez and M. Erdmann, Comput. Phys. Commun. **83**, 171 (1994).
[4] P. Langley, H. Simon, G. Bradshaw, and J. Zytkow, *Scientific Discovery: Computational Explorations of the Creative Processes* (MIT Press, Cambridge, MA, 1987).
[5] H. Baer and W. Long, Comput. Phys. **6**, 24 (1992).
[6] K. G. Wilson, Future Gener. Comput. Syst. **5**, 171 (1989).
[7] P. Fischer and J. M. Zytkow, in *Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems*, edited by Z. Ras, M. Zemankova, and M. Emrich (North Holland, Amsterdam, 1990), pp. 362–370.
[8] H. Simon, in *Mind and Cosmos*, edited by R. Colodny (University of Pittsburgh Press, Pittsburgh, 1966), pp. 22–40.
[9] R. E. Valdes-Perez, J. M. Zytkow, and H. A. Simon, in *Proceedings of the 11th National Conference on Artificial Intelligence* (AAAI Press, Menlo Park, CA, 1993), pp. 472–478.
[10] R. E. Valdes-Perez, Comput. Phys. Commun. **94**, 25 (1996).
[11] H. Fritzsch, *Quarks: The Stuff of Matter* (Basic Books, New York, 1983).
[12] F. Halzen, *Quarks and Leptons: An Introductory Course in Modern Particle Physics* (Wiley, New York, 1984).
[13] *Particle Properties Data Booklet* (North Holland, Amsterdam, 1988).
[14] H. Harari, Phys. Lett. B **86**, 83 (1979).
[15] R. Jaffe, Phys. Rev. Lett. **38**, 195 (1977).
[16] M. A. Shupe, Phys. Lett. **86B**, 87 (1979).
[17] S. T. Tsou, Nucl. Phys. B **141**, 397 (1978).
[18] The two programs described below (named GELL-MANN and YUVAL) are available on request from J.M. Żytkow (zytkow@cs.twsu.edu) and R.E. Valdés-Pérez (valdes@cs.cmu.edu), respectively.
[19] Ulrich Müller (private communication).