Heuristics for Systematic Elucidation of Reaction Pathways

Raúl E. Valdés-Pérez

Department of Computer Science and Center for Light Microscope Imaging and Biotechnology,
Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Received February 28, 1994*

MECHEM is a computer aid for the elucidation of reaction pathways or mechanisms that has reached a promising level of competence at its task: finding the simplest reaction pathways consistent with given experimental evidence and background chemical theory. The program has been applied to complex multistep reactions of hydrocarbon catalysis, with interesting results. However, the program could not reach its current level without several new heuristics. This article describes the heuristics, of which the most important is a divide-and-conquer strategy. Although our focus is on systematic methods, it is noteworthy that standard references on reaction mechanism elucidation do not mention these heuristics.

INTRODUCTION

Given typical experimental facts about a reaction, the a-priori space of reaction pathways that could explain these facts, whether simply or not, is very large. Since explicitly considering all imaginable pathways is infeasible, in practice it is necessary to rely on assumptions and simplifications in order to arrive at a credible, candidate pathway. This is true whether a pathway is inferred by man or by formal means: the space is too large to examine exhaustively.

MECHEM is a computer program, developed over the past 6 years, that serves as an aid for elucidating chemical reaction mechanisms or pathways. The strategy is to combine the knowledge and intuition of the human chemist with the capability for search and calculation of the machine. The program is designed to search exhaustively for the *simplest* pathways that are consistent with experimental facts, ad-hoc assumptions, and background chemical theory. What to do with these pathways is left to the user or to other techniques. Simplicity within the program is measured in terms of the number of steps and species appearing in a pathway. It is not practical to examine manally all conceivable pathways up to a given, nontrivial degree of simplicity, but perhaps surprisingly it has proved possible to do so by computation. This fact does not contradict the previous conclusion about the inconsiderably large space of pathways, since the following heuristic is typically used by MECHEM: disregard all pathways that are more complex than the simplest possible pathways consistent with the given constraints. The pathway-generation algorithm in MECHEM automatically incorporates this heuristic.[1]

However, even with the simplicity heuristic (supplemented of course by the available experimental, theoretical, and ad-hoc constraints on plausible pathways), MECHEM cannot within reasonable time find the simplest pathways for multistep reactions of a challenging degree of complexity, e.g., that involve "conjecturing" six unseen intermediates and twelve elementary steps. The aim of this article is to describe other heuristics developed for use in MECHEM that have allowed the program to handle challenging complex reactions. Some of the heuristics to be described here are logically redundant, in the sense that their use should not affect the program's conclusions (assuming that enough time were available to finish the computation), since they prune parts of the search that would eventually lead to a dead end anyway. However, these heuristics yield great savings in computation and render feasible many problems that otherwise would be beyond the reach of even the fastest computers.

We proceed by giving an overview of MECHEM in order to provide some further context for the application of the heuristics; this overview is similar to those given elsewhere and is included here only for the convenience of readers who are unfamiliar with the program. Next, there is a brief introduction to the concepts of heuristic search, which has found much application within computers in chemistry. Then, the article introduces a systematic classification of heuristics pertinent to pathway elucidation. Finally, there is an analysis of five powerful heuristics in MECHEM and empirical reports on how the program behaves with and without the heuristics.

OVERVIEW OF MECHEM

MECHEM is a computer aid for elucidation of reaction pathways or mechanisms.[1,2] The philosophy underlying the design of the program is to carry out a comprehensive search for the simplest pathways that can account for all of the heterogeneous types of experimental evidence that arise in practice, without making overly restrictive simplifying assumptions. For example, a simplifying assumption *not* made in MECHEM is that all reaction intermediates (or products) are known to the experimentalist/user.

For various types of evidence E, we have tried to design an algorithm to test whether a given pathway P is consistent with E. Sometimes this task has been trivial, but other times it has led to new technical results.[3-5] Our belief is that it should be possible to find an automatic procedure to test most P against a given E; otherwise E is too complex or ill-defined to serve as a reliable constraint on reaction pathways.

The key technical discovery underlying the program was an algorithm[1] to generate reaction pathways nonredundantly[6] in order of simplicity, where simplicity is measured in terms of the number of reaction steps and total number of species appearing in a pathway. The algorithm is able to "conjecture" unseen species in a novel way: by introducing wild cards such as X, Y, Z, etc., for which molecular formulas are inferred by balancing all the steps in which the wild cards occur. Then, another algorithm[7] infers a set of possible molecular structures for each wild card, given its formula and the context in which it is formed in a pathway. Fewer conjectured species are tried first; i.e., $k + 1$ conjectures are made only when k conjectures have proved inadequate to satisfy all the constraints.

* Abstract published in *Advance ACS Abstracts*, June 1, 1994.

0095-2338/94/1634-0976$04.50/0

The general spirit is to consider the entire "space" of possible hypotheses, which in the present case consists of all possible sets of steps *reactants → products* having at most two reactants and at most two products. Given a set of possible reactants and a second set of possible products (including wild cards), MECHEM generates reaction steps exhaustively by forming all instances of the (schematic) steps $A + B \rightarrow C + D$, $A \rightarrow C + D$, etc. Thus, starting with the empty pathway, the program first forms all possible first steps, in which the reactants can only be drawn from the starting materials. Then, for each first step R_1, the program considers all possible second steps R_2, where the reactants of any R_2 must be drawn from the starting materials or from the product(s) of R_1. A recursion of this procedure generates a tree of pathways, in which every extra level in the tree corresponds to another reaction step.

Within this tree of candidate pathways, one searches for the simplest pathways (i.e., set of elementary steps) consistent with the constraints derived from theory and from experiments with the reaction. If there were no constraints at all from experiments, then the simplest pathway would be the empty set. If at least one product of the reaction is known, then any acceptable pathway must explain how that product is formed. In general, any fact must receive an explanation, and the more constraining facts are known, the more credible are the pathways that are found. Whether such a systematic approach will succeed in chemistry is best judged over time.

The program is used interactively: after a run on a set of constraints, typically the user will, drawing on his knowledge and intuition, object to certain aspects of the reported pathways. After articulating the objection, new constraints are formulated and the program is restarted with the new constraints. This interactive process continues until the user is satisfied with the results. The program cannot handle all reactions, but a significant class of moderately complex chemistry is within its scope.

A recent result achieved with MECHEM was to find a plausible alternative pathway for the metal-catalyzed hydrogenolysis of ethane.[8] That reaction has been studied for over 28 years, and several mechanisms have been proposed. MECHEM found a new pathway of comparable simplicity which involves hydrogen transfer between adsorbed species, rather than the successive hydrogenation which the accepted mechanisms had featured. All the chemical examples in this article will be taken from catalytic chemistry, since that is the field of most promising application.

The approach in MECHEM is logic-oriented, following the distinctions made by Ugi et al.,[9] but differs from the programs developed by the Munich project headed by Ugi as follows. The Munich programs, some of which address elucidation of reaction mechanisms, are based on a "constitution-oriented algebra" of reaction and bond matrices and carry out their top-level reasoning within the space of bond configurations. MECHEM, on the other hand, carries out its top-level reasoning in the space of pathways: the space of molecular structures is present in the program but is subordinate to its search in the space of pathways. Thus, MECHEM will generate a reaction step before testing its structural soundness, whereas the RAIN program[10] generates steps by employing structural transforms. The advantage, in MECHEM's case, is that many constraints can be applied to rule out a step $A + B \rightarrow C + D$, or a sequence of such steps, without delving into the structural transformations involved, and testing these constraints tends to be inexpensive. For example, the constraint that one species not be prerequisite for another can be tested at the *pathway* or *reaction network* level, without regard for the mechanistic aspects therein.

Finally, MECHEM complements the systematic methods for microkinetic analysis in heterogeneous catalysis developed recently by Dumesic, Rudd, and colleagues,[11] which address the problem of determining the rate constants of the steps within a given serviceable reaction mechanism. The current program addresses the latter, initial problem of finding a mechanism.

HEURISTIC SEARCH

The basic method used by MECHEM is heuristic search, which involves exploring a (typically large) space of possibilities selectively by the use of heuristics. Heuristics are bits of knowledge, or rules of thumb, that can serve to focus attention on one part of the space versus another. Heuristics can be strictly sound or of only modest reliability; the important issue is whether they serve to explore one region of the space in preference to another, presumably inferior region. An instructive illustration of the concept of heuristic search is the task faced by the familiar TV homicide detective. The detective uses powerful heuristics to investigate a small set of suspects from the logically much larger set of human beings on the planet: Who had a motive? Who last saw the victim?

The theory of heuristic search[12] asserts a broad scope for this concept, aspiring to account even for scientific reasoning,[13,14] as well as to provide a guide for designing computer programs that carry out significant tasks in science.[15] Chemistry has seen a successful application of heuristic search to the problem of automated synthesis. For example, Corey et al.[16] have described the design of the LHASA program as drawing "heavily upon techniques more familiar to the computer scientist than the chemist. The overall process is based on the principles of 'heuristic search'...". Most of the automatic synthesis programs are based on a heuristic search within a tree of syntheses, e.g., SECS,[17] SYNCHEM,[18] SYNGEN,[19] and many others. CAMEO,[20] on the other hand, seems to rely less on heuristic search than on knowledge about specific reactions.

The concept of "heuristic" has received several interpretations, some of which strain to distinguish sound heuristics from uncertain rules of thumb. In this article we speak of heuristics as any piece of knowledge that serves to rule out conceivable pathways. Thus construed, it is considered a heuristic both to disregard elementary reaction steps that do not balance (otherwise, where did the mass go?), as well as to prefer simple pathways over needlessly complex ones. The scope for our use here of the term "heuristics" will include basic laws of chemistry such as balance, since these laws serve to reject regions of the search space. Whether based on chemical law or nonlaw, the heuristics uniformly serve as tools of inference and, hence, will be treated uniformly here.

MECHEM shares with the well-known DENDRAL program[21] a search regimen of constrained generate-and-test, in which combinatorial structures are generated piece by piece, with various constraints applying at various points within the generator: at early points, where small combinatorial structures are being generated, or at later points where nearly complete structures are considered. For example, in DENDRAL's task of elucidation of molecular structure, a constraint that carbon has a valence of 4 serves to reject any molecular fragment in which a carbon is linked to five atoms, without waiting to see how that fragment is filled in further to make up a whole molecule having the known formulas. A general principle of generate-and-test search regimens is that it pays to incorporate constraints very early in the generator, since

Table 1. Classification of Example Constraints on Reaction Pathways

object	examples of constraint	object	examples of constraint
formula	unsaturation	step	balance
structure	valence		mechanism
reactant	inertness		free energy
coreactants	phase	set of steps	precursors
product	formability	complete pathway	concentrations, catalysis, overall stoichiometry
coproducts	unsaturation		

doing so will forestall subsequent elaborations of a partial structure which are doomed to be rejected anyway.

HEURISTICS FOR PATHWAY ELUCIDATION

There are several references on pathway or mechanism elucidation that are instructive to the point of providing helpful hints to develop skill at the task. Some years ago, for example, Edwards, Greene, and Ross in the *Journal of Chemical Education* provided several kinetics-based rules of thumb for inferring aspects of reaction mechanisms.[22] A entire recent book by Miller also targets the development of skill, as revealed by its title: Writing Reaction Mechanisms in Organic Chemistry.[23] Another book by Carpenter is more oriented toward experimental techniques but also clearly addresses the problem of elucidation.[24] Other often-cited references include a chapter by Margerison on the treatment of experimental data in chemical kinetics.[25] The heuristics described in this article do not appear in these sources. However, we will propose that they can be usefully applied even outside the context of computer-aided elucidation, although the latter is the main focus here.

Systematic Classification of Pathway Heuristics. Table 1 proposes a systematic way to classify the rich sources of constraint that bear on chemical reaction pathways. This systematic classification is reflected in MECHEM, since for each row of the table there is a corresponding subroutine in the program: **test-formula**, **test-step**, etc.

Let us examine the plausibility tests of molecular formulas. The very first constraint that needs to be expressed is that formulas are positive lists of numbers; e.g., having a negative number of oxygens is not plausible. More sophisticated tests are also available: one can calculate the degree of unsaturation of a formula using the standard algebraic expression, and if the result comes out negative, the formula is rejected as implausible. For example, a formula CH_5 might be heuristically rejected solely on the basis of the degree of unsaturation (CH_5 gives $-^1/_2$), without the need for any reasoning about molecular structure.

Other constraint categories in Table 1 bear on higher aggregates within the hierarchy of pathway objects. For example, in reactions of heterogeneous catalysis, one might have a constraint on coreactants that prohibits any reaction between two gas-phase species. Below, we discuss in detail a heuristic constraint on coproducts (or coreactants) of a step which prohibits two fully-saturated species X and Y from appearing in a step as *reactants* $\rightarrow$ X + Y, or as X + Y $\rightarrow$ *products*.

Some constraints on pathway objects represent generally-applicable constraints that belong to background chemical theory, e.g., the requirement that steps balance is present in most of chemistry. Other constraints are ad hoc, in the sense that they pertain only to the specific reaction under study. For example, in a reaction of heterogeneous catalysis, one might assume, on the basis of experimental evidence, that no elementary reactions occur solely in the gas phase; this constraint is not generally true, of course. In MECHEM there are also subroutines **test-formula-ad-hoc**, **test-step-ad-hoc**, etc., which provide a convenient means for inserting reaction-specific assumptions.

One value of the classification is that it offers a systematic way not only to insert knowledge into a program such as MECHEM but also to look for other applicable constraints. For example, the *no-two-saturated-species* heuristic discussed below was invented by focusing on what unsaturation-inspired constraint could be formulated on reactants or products to reduce the combinatorial search. The classification also provides a systematic way to "interview" the experimentalist about possible constraints on his reaction, although this is not yet, perhaps, a familiar way of thinking about reactions.

Heuristics in MECHEM. The most basic starting point for the design of the pathway generator in MECHEM involves a powerful heuristic: pay close attention to the experimentally observed products and intermediates, and ask how they might be derived from the starting materials. Much discussion of reaction-mechanism elucidation emphasizes instead the issue of prediction, of what reactions might occur between certain reactants. MECHEM's behavior on complex, practical reactions shows that the previous heuristic, together with the other techniques and heuristics in the program, can result in competent performance without any capability to *predict*, given two species A and B, what is likely to occur. This result is perhaps counterintuitive.

A second basic, powerful heuristic in MECHEM is *simplicity*, which serves to focus attention on pathways in which fewer steps and total species appear. This heuristic is integral to the main search algorithm underlying the program and has been reported in detail previously.[1] The basic effect is to generate more complex pathways only when simpler ones prove inadequate. Within this class of "formal" heuristics is a canonical representation of pathways[6] that serves to disregard any partially-built pathway that violates the canonical ordering. Canonicalization heuristics in chemistry date at least to the DENDRAL program,[21] although that program had a canon for molecular structures, rather than a canon for multistep pathways as in the present case.

Other heuristics in MECHEM concern obvious chemical constraints that are general in applicability, such as reaction balance, molecularity, constraints on atomic valences, or a threshold on the number of bond changes per elementary step.[26] Still another class of heuristics derive from experiments on a particular reaction and may require, for example, that a given starting material catalyze the reaction,[4] that one intermediate not be a precursor of another, or that the overall stoichiometry be some specific expression.

These heuristics have been adequately described in the previous articles on the various algorithms within MECHEM. The present article describes five unreported heuristics that contribute significant power to the program. The most important of these is a divide-and-conquer heuristic without which the practical scope of the program would be greatly reduced. For example, the application of MECHEM to the hydrogenolysis of ethane in heterogeneous catalysis, which resulted in a novel, plausible pathway of simplicity equal to an accepted version, could not have succeeded without divide-and-conquer for reasons of combinatorial complexity. The following discusses these various new heuristics.

Divide and Conquer. The practical problem of pathway elucidation is formalized as follows. Given the starting materials SM, any observed intermediates or products P, and constraints C, i.e.,

$$\mathrm{SM} \rightsquigarrow \mathrm{P} \quad \text{and} \quad \mathrm{C} \text{ is required}$$

find one or more pathways that account for the formation of P from SM and that are consistent with C, i.e., make C true. A divide-and-conquer reformulation of the problem depends on partitioning the set of products or intermediates P into P_1 and P_2 such that $P_1 \cup P_2 = P$ thus:

stage 1 $\mathrm{SM} \rightsquigarrow P_1$ and C is required

stage 2 $\mathrm{SM} \cup P_1 \cup \text{products}(\text{pathway}_1) \rightsquigarrow P_2$ and C is required

Having thus divided the problem into two stages, one first conquers the subproblem of explaining $\mathrm{SM} \rightsquigarrow P_1$. Then, for every partial pathway *pathway₁* found in stage 1, one augments the original starting materials with all the species formed in *pathway₁* and conquers the second subproblem. Each partial pathway found in this stage 2 is appended to *pathway₁* to give an overall pathway that fulfills the original task of explaining $\mathrm{SM} \rightsquigarrow \mathrm{P}$. In all cases, C must be true, except when a constraint in C pertains only to a completed pathway, e.g., when a specified overall stoichiometry appears as a constraint in C.

We illustrate the divide-and-conquer heuristic with an ethane hydrogenolysis reaction from heterogeneous catalysis (we will preserve the conventional "+" notation even though the "$\rightsquigarrow$" formalism above refers strictly to *sets* of species):

ethane + H_2 + M(catalyst) $\rightsquigarrow$ MCH_2MCH_2 + MCHMCH + methane

The overall stoichiometry for this reaction is constrained to be

$$\text{ethane} + H_2 \rightarrow 2(\text{methane})$$

which implies that MCH_2MCH_2 and MCHMCH are intermediates. Hence, a natural partition of the problem is the following:

stage 1 ethane + H_2 + M $\rightsquigarrow$ MCH_2MCH_2 + MCHMCH

stage 2 ethane + H_2 + M +
MCH_2MCH_2 + MCHMCH + products(pathway$_1$) $\rightsquigarrow$ methane

After splicing together the partial pathways of the two stages, any overall pathway that cannot account for the specified overall stoichiometry of ethane + $H_2 \rightarrow$ 2(methane) is rejected. If none is satisfactory, then more complex partial pathways are sought in either of the two stages.

In some cases, one may prefer to assume, for example, that the H_2 of stage 1 is no longer available as a starting material in stage 2, thus reducing the complexity of the problem in stage 2. When applying MECHEM to this ethane hydrogenolysis reaction, an initial step generated by the program is $H_2 + 2(M) \rightarrow 2MH$, so that the chemisorbed MH takes over the role of H_2 in the reaction.

What is the purpose here of the divide-and-conquer heuristic, which is very common in the design of algorithms? Its purpose is to reduce the size of a problem, and thus the computational cost needed to solve it. We illustrate this as follows. Let us suppose the computation time needed to solve a problem of size L increases exponentially with L, so that time $\propto L^{\alpha}$. We further suppose that L can be partitioned into L_1 and L_2 and solved separately and that merging the local solutions into a global solution is straightforward. Then, the computation time for the partitioned problem is time $\propto L_1^{\alpha} + L_2^{\alpha}$ which is much less than L^{α} if $\alpha \gg 1$.

In MECHEM, the computing time is determined most strongly by the number of unseen species that the program must "conjecture" in order to find a satisfactory pathway. On ordinary workstations, the program has been unable to finish on problems that involve introducing more than four unseen species.[27] If one can divide a reaction problem into two (or more) pieces, then the program can handle reactions in which eight or more unseen species (e.g., intermediates) will need to be conjectured. This number is high enough to include many complex reactions of scientific or engineering interest. For example, on the ethane hydrogenolysis reaction, MECHEM needed six unseen species to find the simplest satisfactory pathways; one of the pathways was identical to a proposal by Gudkov et al.[30] The program also found a seemingly novel pathway which involved hydrogen transfer between adsorbates, rather than the successive hydrogen abstractions and additions featured by the Gudkov mechanism.[8] The program could never find these pathways without the divide-and-conquer heuristic, subject to the previous qualifications.[27]

A partition of a reaction problem into intermediates and stoichiometric products, as discussed above, is not the only sound way to divide a problem into pieces. Let us consider another catalytic reaction, the partial oxidation of methane:

methane + O_2 + M(catalyst) $\rightsquigarrow$
CH_3 + ethane + methanol + water + CO + CO_2 + formaldehyde + H_2

Again, MECHEM cannot currently handle this problem without dividing it into two pieces. The following partition leads to an interesting result:

stage 1 methane + O_2 + M $\rightsquigarrow$ CH_3 + ethane + methanol + water + H_2

stage 2 products(pathway$_1$) $\rightsquigarrow$ CO + CO_2 + formaldehyde

The basis for this partition is a constraint that elementary steps increase or conserve the oxidation numbers while going from reactants to products. Hence, the products having the highest oxidation numbers (CO, CO_2, and formaldehyde) were assigned to stage 2. Here is one of the simplest pathways found by the program[28] (the six "conjectured" species are shown between angle brackets "⟨ ⟩":

$$CH_4 + M \rightarrow \langle MCH_3 \rangle + \langle H \rangle \tag{1}$$

$$\langle MCH_3 \rangle \rightarrow M + CH_3 \tag{2}$$

$$O_2 + \langle MCH_3 \rangle \rightarrow \langle MCH_2OOH \rangle \tag{3}$$

$$2\langle H \rangle \rightarrow H_2 \tag{4}$$

$$2(CH_3) \rightarrow CH_3CH_3 \tag{5}$$

$$\langle H \rangle + \langle MCH_2OOH \rangle \rightarrow H_2O + \langle MCH_2O \rangle \tag{6}$$

$$CH_3 + \langle MCH_2OOH \rangle \rightarrow CH_3OH + \langle MCH_2O \rangle \tag{7}$$

$$\langle MCH_2O \rangle \rightarrow M + CH_2O \tag{8}$$

$$\langle MCH_2OOH \rangle \rightarrow \langle H \rangle + \langle MCH_2OO \rangle \quad (9)$$

$$CH_2O + \langle MCH_2OO \rangle \rightarrow \langle MCH_2OOH \rangle + \langle HCO \rangle \quad (10)$$

$$\langle MCH_2OO \rangle + \langle HCO \rangle \rightarrow \langle MCH_2OOH \rangle + CO \quad (11)$$

$$CO + \langle MCH_2OO \rangle \rightarrow \langle MCH_2O \rangle + CO_2 \quad (12)$$

The divide-and-conquer heuristic is crucial for the success of systematic, comprehensive methods for elucidation of reaction pathways. Without it, only reactions of rather modest complexity could be handled. While the heuristic was developed for computer-based methods, it seems a powerful heuristic also for the chemist, and it is likely that model builders use the heuristic at least subconsciously. This heuristic is not discussed in the references cited previously. Finally, this is the only heuristic of this article that does not fit into the classification of Table 1; the reason is that it is not a constraint on solutions but a way to reformulate an elucidation problem.

Does the divide-and-conquer heuristic preserve any guarantee that the simplest pathways will still be found? We consider the elucidation problem SM $\rightsquigarrow$ P. Let us assume that we know the correct pathway, and that no more than eight species will need to be conjectured to formulate the pathway. Now, the pathway steps can be sorted according to, for example, a canonical order.[6] Then, starting at the first step, one collects steps one at a time into a bag B until adding a step k would imply adding a fifth conjectured species to B. At that point, the $k - 1$ steps in B constitute a solution to the divided elucidation problem SM $\rightsquigarrow$ P_1 in which the P_1 are those products in P that appear anywhere in B. The second stage of the problem deals with accounting for the products in $P - P_1$. Therefore, we see that there exists a partition of the elucidation problem that will still arrive at the correct pathway, unless of course a simpler pathway can account for the known constraints. Trying all the partitions will guarantee finding the correct (or simpler) pathway. If one employs certain simplifications to avoid examining all possibilities, then one loses the guarantee. This was done above by partitioning the observed products according to oxidation number. It is unclear whether a partition based on stoichiometric intermediates versus products still preserves any guarantee.

Go Forward. If the goal were only to connect somehow the starting materials to the observed products via a pathway subject to no constraints on sets of steps or the overall pathway and if one were building a pathway by adding steps one at a time, then one could disregard any ith step of the form *reactants* $\rightarrow P_i$ in which all the species in P_i were either starting materials or were formed by an earlier step, i.e., any step from 1 to $i - 1$. We have named this the *go-forward* heuristic. For example, let us consider the following partial pathway actually generated by MECHEM in its search:

$$CH_4 + M \rightarrow CH_3 + \langle MH \rangle$$

$$O_2 + \langle MH \rangle \rightarrow \langle MOOH \rangle$$

$$CH_3 + \langle MOOH \rangle \rightarrow M + \langle CH_3OOH \rangle$$

$$\langle MH \rangle + \langle CH_3OOH \rangle \rightarrow CH_4 + \langle MOOH \rangle$$

The last step violates the go-forward heuristic, since its two products are both present earlier in the pathway: CH_4 as a starting material and MOOH as a conjectured intermediate. Thus, this point in the search is cut off, although other fourth steps will be considered. The intuitive justification is that adding such steps does not advance the assumed goal of explaining how all of the observed products are formed, since the step's products were, by assumption, already present earlier in the pathway.

One can devise the following applicability conditions for the go-forward heuristic: If the only available constraint on multiple steps (i.e., a partial or complete pathway) is the requirement to account for observed products, then the go-forward heuristic can be used unreservedly to find the simplest pathways.

Now, what of the usual cases where further pathway constraints are known, for example, where one knows the overall stoichiometry of the reaction? In such cases, the go-forward heuristic can still be used with advantage, but after finding a pathway to account for all of the required products, one carries out a second stage as follows:

1. Let the set S consist of all the pathway species, i.e., starting materials, observed products (whether final or intermediate), and conjectured species.
2. Drawing only on S, form all possible single steps of the form

$$r_1 + ... + r_j \rightarrow p_1 + ... + p_k \quad (j, k \leq 2)$$

   that violate no available constraint, the most elementary of which is, of course, that the step must be balanced.
3. Augment the pathway with the newly generated steps; then test whether the remaining pathway constraints (e.g., overall stoichiometry) can be satisfied by the augmented pathway. If not, then the unaugmented pathway is a dead end and will not be part of a solution; further consideration of it can cease.

Go-forward appears to be a variant of a heuristic used in conjunction with sequential interchanges of functionality in the LHASA program.[31] However, the heuristic's justification seems different in the case of reaction elucidation versus reaction synthesis.

It is useful to distinguish between two types of constraint on the overall reaction pathways. In one type, a pathway P that violates a given constraint cannot be made to satisfy it by adding more steps to P that involve only species already appearing in P. In the second type, a violated constraint can be satisfied by adding one (or more) steps. We propose the term 'stable' to distinguish the two types: a constraint is stable (unstable) if adding a step cannot (can) serve to change the constraint's status from 'violated' to 'satisfied'.

We illustrate the concept of constraint stability with two examples:

The constraint of overall stoichiometry is unstable. For example, an overall stoichiometry of A + 2B $\rightarrow$ 2C cannot be explained by a pathway consisting of the two steps A $\rightarrow$ X + Y and B + X $\rightarrow$ C. However, augmenting this pathway with the further step Y $\rightarrow$ X does satisfy the overall stoichiometry; i.e., there exist stoichiometric numbers that imply the overall stoichiometry A + 2B $\rightarrow$ 2C.

The constraint "species X is a precursor of species Y" is also unstable, since it may be false for a pathway P but may become true by adding a step such as X $\rightarrow$ Y + Z, where the species X, Y, and Z already appear in P.

Not all constraints are unstable. For example, the illegal presence of an unbalanced pathway step cannot be rectified by adding more steps to the pathway. Moreover, not all "global" constraints are necessarily unstable; the hypothetical constraint "the pathway cannot contain more than N steps" cannot be satisfied by adding more steps. However, in our experience, it seems that most practical global constraints (i.e., that pertain to pathways as a whole) are unstable in the above sense.

The value of the constraint stability concept is as follows. The go-forward heuristic can be used first to account for the presence of the required products, while applying any available stable constraints. Then, extra steps can be appended, whose earlier introduction would have violated the go-forward heuristic. Finally, any unstable constraints should be applied to the thus-augmented pathway. The aim of proceeding in this rather complicated manner is to save computation: the go-forward heuristic reduces considerably the combinatorics of pathway generation, as shown by the experimental tests reported later in this article. According to the classification of Table 1, the go-forward heuristic is a constraint on sets of steps.

No Transmutation. During a systematic search for pathways, one may generate partial results such as this:

$$CH_4 + \text{catalyst} \rightarrow X + Y$$

$$\text{catalyst} + X \rightarrow CH_3 + Z$$

$$2(CH_3) \rightarrow CH_3CH_3$$

$$CH_3CH_3 + Z \rightarrow CH_3OH + Y$$

where X, Y, and Z are wild card variables. These steps are clearly unrealizable, since there is an oxygen-containing product (CH_3OH), but no oxygen ever enters into the reaction (assume that the catalyst cannot contribute oxygen). When, as is often true, the starting materials do not all contain the same elements, many such partial pathways will be pursued at some length during the search. Eventually, all the extensions of these will be rejected after enough further constraint is added (by adding more reaction steps) to determine by simple matrix algebra the formulas for the wild-card variables. In the above example, though, enough constraint is available to immediately infer unique formulas for the variables:

$$X = C_{1/2}H_{5/2}O_{1/2}$$

$$Y = C_{1/2}H_{3/2}O_{-1/2}M$$

$$Z = C_{-1/2}H_{-1/2}O_{1/2}M$$

and these absurd values would lead to rejection of the steps anyway. However, in the case of shorter partial pathways such as

$$CH_4 + \text{catalyst} \rightarrow X + Y$$

$$\text{catalyst} + X \rightarrow CH_3OH + Z$$

considerable added search will be needed before unique values are inferred for the variables. Yet, it is clear that these steps are unsatisfiable as well.

It is worthwhile to introduce a simple heuristic to intercept such cases, even though they will eventually lead to dead ends anyway; the goal is to save needless computation. We have used the following heuristic, which applies whenever a new step is added to a partial pathway:

1. For all the current steps, collect all the reactants that have a known molecular formula (i.e., ignore variables); then sum their formulas.
2. Collect and sum the formulas for the products of the last step. (If the products consist only of undetermined variables, then the heuristic does not apply.)
3. If the products-sum has a nonzero coefficient where the reactants-sum has zero, then reject the new step.

A stronger heuristic which instead had this test: "If the products-sum has a coefficient greater than the corresponding coefficient in the reactants-sum, then reject the new step" would fail, since it rejects perfectly good steps such as

$$CH_4 + \text{catalyst} \rightarrow X + Y$$

$$X + X \rightarrow CH_3CH_3$$

One could use a more thorough heuristic that is based on finding satisfactory formula assignments to the variables. Since in many cases during the search there is not enough constraint to infer unique values via a straightforward algorithm such as Gaussian elimination, a more thorough heuristic would incur another search of its own. Our preliminary tests have not shown this tactic to be worthwhile, although it is available in MECHEM.

According to the classification of Table 1, the *no-transmutation* heuristic is a constraint on sets of steps, just like the previous go-forward heuristic. Although the truth of the no-transmutation heuristic is obvious, the need for it, as well as its power, is not. MECHEM will, of course, never report a pathway that involves transmutation of the elements, so logically the heuristic is superfluous. In practice, though, a MECHEM-style search will waste much time on steps that imply transmutation unless this heuristic is used.

Need All Starting Materials. In practical cases of pathway elucidation, it is usually true that the starting materials are actually needed for formation of the products, or to be exact, that there is at least one observed product that will not form if one of the starting materials is omitted. This immediately suggests the following heuristic: never append to a partial pathway a step that forms a starting material that has not reacted earlier in the pathway. The justification is that adding such a prohibited step would imply that the starting material in question is not strictly needed, since the other starting species can generate some of it themselves. Of course, this is a constraint that should be decided experimentally when there is any reason to doubt its veracity.

One might be tempted to strengthen the heuristic by preventing all steps from producing any starting material, whether already consumed earlier in the pathway or not. On an ad-hoc basis, the strengthened heuristic might be justifiable, but its unqualified use would rule out, for example, catalytic pathways in which a reacting catalyst must be regenerated later in the pathway.

The *need-all-starting-materials* heuristic is a constraint on sets of steps, according to Table 1.

No Two Saturated Species. MECHEM incorporates a constraint that requires all plausible elementary steps to have at most *N* changes in bonding (whether cleavage or breakage), where *N* is an adjustable parameter that is 3 by default.[26] Any generated step that exceeds the *N* threshold is rejected.

With the default threshold of 3 on the number of bond changes, it turns out that one can develop a heuristic to reject any reaction between two saturated species (this heuristic assumes, as does MECHEM, that every elementary step consists of at most two reactants and at most two products).

For example, the elementary step

$$CH_3CH_3 + H_2 \rightarrow \text{products}$$

can be rejected without carrying out detailed mechanistic reasoning about a fully-specified instance of the step, indeed without even knowing what the products may be. The justification is as follows.[29]

Since both reactants are saturated, there is no place to form a bond on either, so a reaction must proceed by breaking a bond within each of the two reactants, thus yielding four fragments. Since, by assumption, only one bond change remains available (in this case a formation) to carry out the step, at best the four fragments can be reduced to three. Hence, it is not possible to formulate any step

$$CH_3CH_3 + H_2 \rightarrow X + Y \text{ (or} \rightarrow X)$$

without exceeding the default bond-change threshold of three, regardless of the identities of X and Y. If this heuristic were not applied, any such step would eventually be rejected, but only after generating a large number of subsequent steps corresponding to different products, and then verifying that the minimum bond changes exceed 3, all of which implies must wasted computation that could be preempted by application of the no-two-saturated-species heuristic earlier in the generation, even before the products of such a step are generated.

In the general case, the default heuristic will reject some concerted reactions. However, changes to the multiplicity of bonds (e.g., from double to single or vice versa) are not counted toward the threshold on bond changes,[7] which counts only changes that affect chemical-graph connectivity. Hence, cycloadditions such as the Diels–Alder reaction are not ruled out. If even this cautious threshold is unwarranted in a specific application, then the threshold can always be set to four bond changes instead of three. Future developments of the program should include context-dependent refinements to this uniform approach.

In the above partial oxidative reaction of methane, five of the species given as input are saturated: methane, ethane, water, methanol, and H_2. Any generated bimolecular reactions among them are immediately rejected.

The previous three heuristics were examples of constraints on sets of steps. The no-two-saturated-species heuristic, unlike the others, is a constraint on coreactants as well as on coproducts, since its justification does not depend on the direction of the elementary step.

EXPERIMENTAL TEST OF HEURISTIC POWER

To determine the power of the main heuristics of this article, at least when applied by MECHEM, we ran the computer program with and without the heuristics to determine the speed-up and to verify that the same final answers are obtained. The reaction used was the first stage of the above partial oxidation of methane:

$$\text{methane} + O_2 + \text{catalyst} \rightsquigarrow CH_3 + \text{ethane} + \text{methanol} + \text{water}$$

subject to several additional constraints concerning the catalyst and the possible elementary steps. Note that without the divide-and-conquer heuristic, MECHEM could not presently handle the full overall reaction within a practical time, since we have never observed the program to run to completion on any reaction needing five or more conjectured species. The tests here are concerned with the other four heuristics; the power of divide-and-conquer should be clear.

Table 2 shows the results of running MECHEM with and without the heuristics on a Silicon Graphics Indigo workstation. Seven first-stage pathways are found, each of which contains seven steps and four conjectured species. Concerted use of the four heuristics reduces the computing time by 83% from 19.0 to 3.3 h. Of course, the divide-and-conquer heuristic was already applied to formulate the problem reflected in the table, so that the combined effect of the five heuristics of this article is to convert an impracticably long computation to a problem soluble in several workstation hours.

Table 2. Timing Results of the First Stage of the Partial Oxidation of Methane

experiment	timing results (h)
using all heuristics	3.3
without no-two-saturated-species	3.9
without need-all-starting-materials	3.4
without no-transmutation	6.5
without go-forward	9.2
without all four heuristics	19.0

The timings imply that, individually, all of the heuristics are valuable except need-all-starting-materials. That is, except for the latter, the absence of any of the heuristics will increase significantly the computing time. On the test reaction, the absence of need-all-starting-materials did not slow the computation appreciably. This is explained by the fact that the three starting materials (CH_4, O_2, and the catalyst M) do not have any "elements" in common, so none of them can be formed with only the remaining starting materials as precursors (which is the condition tested by the heuristic). Of course, this fact does not always hold: for example, we are currently applying MECHEM to a methane-coupling reaction in which CH_4, O_2, and CO_2 are present in the feed. We predict the heuristic to be useful in such common cases.

CONCLUSION

This article reports several heuristics for systematic elucidation of reaction pathways that have found important application within a computer aid named MECHEM. The program is broadly based on the principles of heuristic search, which is a concept that has found significant application previously within the field of computers in chemistry.

The most important of these heuristics is divide-and-conquer, the use of which enables finding by automated means the simplest reaction pathways for quite complex chemical problems such as the partial oxidation of methane in heterogeneous catalysis, which is one currently targeted application of this research. Without the divide-and-conquer heuristic, such reactions could not be handled on even the fastest computers by the methods we have reported previously. Other heuristics contribute to reducing computation time, which is crucial to taming the combinatorial search inherent in the approach. Of course, some heuristics have a greater impact on some problems and less on others, depending on the detailed characteristics of the chemistry.

Some of these heuristics might be of value outside the context of computer-aided elucidation. We suggest, in particular, that the divide-and-conquer heuristic is used at least subconsciously by reaction model builders who pay close attention to the experimentally-observed products, rather than rely mainly on prediction. Several of the references on reaction mechanism elucidation that were cited above do not discuss these heuristics.

ACKNOWLEDGMENT

This work was supported in part by a Science and Technology Center grant from the (USA) National Science Foundation, No. BIR-8920118, a High Performance Computing and Communications grant from the (USA) National Science Foundation, No. ASC-9217091, and by the W. M. Keck Center for advanced training in computational biology at Carnegie Mellon University, the University of Pittsburgh, and the Pittsburgh Supercomputing Center. We thank an anonymous reviewer for helpful comments and suggestions.

REFERENCES AND NOTES

(1) Valdes-Perez, R. E. Algorithm to Generate Reaction Pathways for Computer-Assisted Elucidation. *J. Comput. Chem.* **1992**, *13*, 1079–1088.

(2) Valdes-Perez, R. E. Symbolic Computing on Reaction Pathways. *Tetrahedron Comput. Methodol.* **1990**, *3*, 277–285.

(3) Valdes-Perez, R. E. On the Concept of Stoichiometry of Reaction Mechanisms. *J. Phys. Chem.* **1991**, *95*, 4918–4921.

(4) Valdes-Perez, R. E. A Necessary Condition for Catalysis in Reaction Pathways. *J. Phys. Chem.* **1992**, *96*, 2394–2396.

(5) Valdes-Perez, R. E. A Correspondence between Reaction-Network Equilibria and Boolean Functions. *Chem. Eng. Sci.* **1990**, *45*, 3384–3386.

(6) Valdes-Perez, R. E. A Canonical Representation of Multistep Reactions. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 554–556.

(7) Valdes-Perez, R. E. Algorithm to Infer the Structures of Molecular Formulas within a Reaction Pathway. *J. Comput. Chem.*, in press.

(8) Valdes-Perez, R. E. Human/Computer Interactive Elucidation of Reaction Mechanisms: Application to Catalyzed Hydrogenolysis of Ethane. *Catal. Lett.*, in press.

(9) Ugi, I.; et al. Computer-Assisted Solution of Chemical Problems-The Historical Development and the Present State of the Art of a New Discipline of Chemistry. *Angew. Chem., Int. Ed. Engl.* **1993**, *32*, 201–227.

(10) Fontain, E.; Reitsam, K. The Generation of Reaction Networks with RAIN. 1. The Reaction Generator. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 96–101.

(11) Dumesic, J.; Rudd, D.; Aparicio, L.; Rekoske, J.; Trevino, A. *The Microkinetics of Heterogeneous Catalysis*; American Chemistry Society: Washington, D.C., 1993.

(12) Simon, H. A. The Theory of Problem Solving. *Models of Discovery*; Reidel: Boston, 1977; Chapter 4.3.

(13) Simon, H. A. Scientific Discovery and the Psychology of Problem Solving. In *Mind and Cosmos*; Colodny, R., Ed.; University of Pittsburgh Press: Pittsburgh, PA, 1966; p 22.

(14) Langley, P.; Simon, H. A.; Bradshaw, G.; Zytkow, J. *Scientific Discovery: Computational Explorations of the Creative Processes*; MIT Press: Cambridge, MA, 1987.

(15) Zytkow, J.; Simon, H. A. Normative Systems of Discovery and Logic of Search. *Synthese* **1988**, *74*, 65–90.

(16) Corey, E. J.; Long, A. K.; Rubenstein, S. D. Computer-Assisted Analysis in Organic Synthesis. *Science* **1985**, *228*, 408–418.

(17) Wipke, W. T.; Ouchi, G. I.; Krishnan, S. Simulation and Evaluation of Chemical Synthesis-SECS: An Application of Artificial Intelligence Techniques. *Artif. Intell.* **1978**, *11*, 173–193.

(18) Gelernter, H. K.; et al. Empirical Explorations of SYNCHEM. *Science* **1977**, *197*, 1041–1049.

(19) Hendrickson, J. B.; Toczko, A. G. SYNGEN Program for Synthesis Design: Basic Computing Techniques. *J. Chem. Int. Comput. Sci.* **1989**, *29*, 137–145.

(20) Jorgensen, W. L.; et al. CAMEO: A Program for the Logical Prediction of the Products of Organic Reactions. *Pure Appl. Chem.* **1990**, *62*, 1921–1932.

(21) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project*; McGraw-Hill: New York, 1980.

(22) Edwards, J. O.; Greene, E. F.; Ross, J. From Stoichiometry and Rate Law to Mechanism. *J. Chem. Educ.* **1968**, *45*, 381–385.

(23) Miller, A. *Writing Reaction Mechanisms in Organic Chemistry*; Academic Press: San Diego, 1992.

(24) Carpenter, B. K. *Determination of Organic Reaction Mechanisms*; Wiley: New York, 1984.

(25) Margerison, D. The Treatment of Experimental Data. In *Comprehensive Chemical Kinetics*; Bamford, C. H., Tipper, C. F. H., Eds.; Elsevier: Amsterdam, 1969; Vol. 1.

(26) Valdes-Perez, R. E. Algorithm to Test the Structural Plausibility of a Proposed Elementary Reaction. *J. Comput. Chem.* **1993**, *14*, 1454–1459.

(27) This could change when MECHEM is modified to run on parallel computers or distributed workstations, or if the comprehensive search it carries out is modified to a more selective, incomplete search. Work on these is in progress.

(28) The intent within this article is not to suggest plausible pathways but to illustrate the program's output on certain constraints using the proposed heuristics. Hence, the energetics of the reaction pathways is not examined.

(29) We assume that the parameter that determines the allowed degrees of unsaturation is set to prohibit formulas with negative unsaturation, e.g., the formula C_2H_7 is ruled out.

(30) Gudkov, B. S.; Guczi, L.; Tetenyi, P. Kinetics and Mechanism of Ethane Hydrogenolysis on Silica-Supported Platinum and Platinum-Iron Catalysis. *J. Catal.* **1982**, *74*, 207–215.