ARTICLES

Substructure Searching Methods: Old and New[†]

John M. Barnard

Barnard Chemical Information Ltd., 46 Uppergate Road, Stannington, Sheffield S6 6BX, U.K.

Received September 15, 1992

The first algorithms for chemical substructure search on computer were developed in the 1950s and 1960s and were widely adopted in systems developed through the 1970s and early 1980s. Since the mid-1980s there has been significant activity in the development of novel algorithms, which have enabled rapid searches to be made in very large databases. The principles underlying the "classic" algorithms are described, and these are contrasted with more recent approaches. Brief mention is made of current research work.

THE PROBLEM

Stated at its simplest level, substructure searching is the process of identifying those members of a set of full structures which contain a specified query substructure. In graph-theoretical terms, it involves testing a series of topological graphs for the existence of a subgraph isomorphism with a specified query graph.

A subgraph isomorphism exists if all the nodes (atoms) of one graph (G_Q) can be mapped to a subset of the nodes of the other graph (G_F) in such a way that the edges (bonds) of G_Q simultaneously map to a subset of the edges in G_F. (In other words, if two nodes in G_Q are joined by an edge, then they can be mapped onto two nodes in G_F if and only if the two nodes in G_F are also joined by an edge; this is known as the adjacency condition.) Furthermore, the labels carried by the nodes and edges (atom type and bond type, respectively) must be identical if the nodes or edges are to be mapped to each other.

As has been pointed out many times before,[1-5] testing for subgraph isomorphism is an NP-complete problem.[6,7] That is, it has been shown to be a member of a class of mathematically equivalent problems for which there are no known algorithms whose worst-case time requirements do not rise exponentially with the size of the input (number of atoms in the two graphs being compared in this case). This can be understood by considering the nature of subgraph isomorphism. The "brute force" algorithm involves trying every possible way of mapping each of the n_Q nodes in G_Q onto one of the n_F nodes in G_F ($n_Q \leq n_F$); each of the $n_F!/(n_F - n_Q)!$ possible mappings must then be tested to see if any of them obeys the adjacency condition. Even for very small graphs, the number of possible mappings rapidly becomes unmanageable, and it is clear, especially when one further considers the number of compounds in most modern chemical structure databases, that a brute force algorithm is not likely to be very useful.

The point about NP-complete problems is that, while algorithms can be found in which the *average* time requirements are acceptable (particularly for the very simple graphs that are used to represent chemical structures), the *worst-case* requirements remain exponential in the size of the input (number of nodes for subgraph isomorphism). Even with the brute-force algorithm, it is possible that the first mapping tried will identify a subgraph isomorphism. Indeed, the algorithm can be stopped as soon as a subgraph isomorphism is found (assuming that separate identification of all possible subgraph isomorphisms is not required). However, if there is in fact no subgraph isomorphism, one still needs to try every possible mapping and to ascertain that none of them conforms to the adjacency condition.

The subgraph isomorphism problem is a generalization of the graph isomorphism problem (which involves attempting to establish a mapping between the nodes of two complete graphs), and many algorithms have also been proposed for this special case. It is not definitely known if the graph isomorphism problem is NP-complete, and polynomial–time algorithms have been found for graph isomorphism in certain restricted types of graph,[1,4] such as trees. The subgraph isomorphism problem has been proved to be NP-complete,[6] and it is believed (though not proven) that there can be no polynomial–time algorithms for NP-complete problems.[7]

Finding subgraph isomorphism algorithms which operate with acceptable average time requirements has occupied the attention of developers for more than 30 years and is still the subject of active research; some authors have gone so far as to describe this obsession as a disease.[4,8] Chemical substructure searching is one of the most important practical applications of subgraph isomorphism algorithms, and the special characteristics of chemical structure graphs (differently "colored" nodes and edges, low connectivity of most nodes, different connectivity values present) have enabled several techniques to be used that might not be applicable in general graphs.

There are three main approaches which have been used, often in combination, to reduce the average time required:

1. Use a faster computer
2. Use various heuristics to improve the chances of finding an isomorphism early on or to reject candidates which cannot give rise to isomorphisms without exhaustive testing.
3. Carry out the most time-consuming operations in a pre-processing of the database, which is independent of the query substructure

The first of these is not in fact as flippant as it appears, and given the exponential increase in computer power per unit cost which has taken place in the last three decades, it might even be considered to have been the most popular approach, especially if parallel-processing solutions to the problem are

[†] Presented at a symposium on Structure Searching organized by the Division of Computers in Chemistry at the 204th National Meeting of the American Chemical Society, Washington, DC, Aug 24, 1992.

0095-2338/93/1633-0532$04.00/0

included under this heading. Though most of the techniques used in the second approach were developed quite a long time ago, they still find wide application and continue to be refined and improved. The third approach has found more recent application and lies behind a number of systems developed in the mid-1980s.

BACK-TRACKING

A back-tracking algorithm for establishing a mapping between two structures was first described by Ray and Kirsch in 1957,[9] though algorithms based on the same principle continue to be published,[10,11] and even claimed as new. In its simplest form, the algorithm starts with an arbitrary node Q_1 in the query graph Q and maps it to an arbitrary node F_1 (with the same label, i.e., atom type) in the file structure graph F. It then proceeds to map each neighbor (Q_2, Q_3, etc.) of Q_1 onto an unmapped neighbor (F_2, F_3, etc.) of F_1; if this is successful, the algorithm continues by trying to map each unmapped neighbor of Q_2 onto an unmapped neighbor of F_2 etc., until all the nodes in Q have been mapped, in which case a subgraph isomorphism has been found. If, at any stage, it proves impossible to find a mapping for a node in Q, the algorithm "back-tracks" to the last successfully mapped node of Q and tries an alternative mapping for it. If there are no further alternative unmapped nodes in F onto which the current node in Q might map, the algorithm back-tracks again. If the back-tracking gets back to the original node in Q, Q_1, and all the alternative possible mappings for Q_1 have been tried, the algorithm terminates because there is no subgraph isomorphism.

Back-tracking algorithms impose the adjacency condition by only examining nodes in each graph that are neighbors to those which have already been mapped, when trying to find further mappings. They offer an improvement over brute-force algorithms by not bothering to complete the mapping of all nodes in Q, once it can be seen that those nodes already mapped cannot form part of a subgraph isomorphism. The worst-case time requirements still rise exponentially with the number of nodes, but the design of the algorithm makes it unlikely that the worst case will arise very often.

There are several ways in which the performance of back-tracking algorithms can be improved still further. More detailed labels than simple atom type can be applied to the nodes (for example, labels encoding information about the atom's neighbors) and can be used to reject possible mappings at an early stage. The order in which alternatives are examined can be chosen to maximize the chance of early rejection: for example, if an unusual heteroatom, with a lot of neighbors, in chosen as Q_1, it is unlikely that very many nodes will be mapped successfully (and, thus, potentially need to be unmapped as the algorithm back-tracks) unless there really is a subgraph isomorphism.

PARTITIONING AND RELAXATION

As an alternative, or more usually an adjunct, to back-tracking, a number of partitioning procedures were developed, mainly in the 1960s and 1970s.[1,12-18] These are all based on the division of the nodes of each graph into subsets of potential correspondents, which are then iteratively refined.

The purpose of the partitioning is to reduce the number of possible mappings which must be investigated and is initially done by using some property of the nodes, such as atom type or number of connections; for example, if the query contains a node which is a nitrogen atom with connections to two other atoms, this can only correspond to file structure nodes which are also nitrogen atoms with connections to at least two other atoms. The initial partitioning is then refined by a process of further subdivision. In some cases this may leave certain query atoms without any potential correspondents in the database structure (in which case there cannot be a subgraph isomorphism).

The refinement step is, in many cases, based on a technique known as *relaxation*, in which the description of a node is enhanced by iteratively examining its immediate neighbors; thus, at each iteration, information from more and more distant nodes can be brought into the description of a particular node. Relaxation is a commonly used technique in chemical structure handling, though not often identified by name; the best-known example is the Morgan algorithm[19] for canonical numbering of the atoms in a molecule. In partitioning-based substructure search applications, the enhanced descriptions of the nodes in a partition can be used to subdivide that partition.

Partitioning algorithms used for substructure search are closely related to algorithms for topological symmetry perception in chemical structures, since this essentially involves finding mappings (other than the trivial one) of the nodes of a structure onto themselves (*automorphisms*).[20-22]

Sussenguth's Algorithm.[12] This was the first-published partitioning algorithm for subgraph isomorphism, though a restricted version of it, for graph isomorphism in directed graphs, had earlier been proposed independently by Unger.[23] In the algorithm, pairs of corresponding sets of nodes are first generated on the basis of various properties of the individual nodes (node value, number of connections, bond types, etc); the intersections of these sets are then used to partition them further. A "connectivity property" is next applied, in which further pairs of sets are generated by examining the sets of correspondents for the neighbors of each node; this stage is essentially a relaxation process.

When no additonal sets are generated by the application of the connectivity property, the algorithm returns to the partition step, and the two are iterated until one of three possible situations arises. In the first, the pairs of sets identify a unique isomorphism, while in the second, the absence of sufficient potential correspondents for one subset of the query nodes leads to the conclusion that no isomorphism exists; in both these cases the algorithm terminates immediately. In the third situation, no further partitions are possible, but those which exist are insufficient to identify a unique isomorphism or to determine that none exists; in this case, the algorithm resorts to conventional back-tracking to resolve the question, though this is required only very occasionally.

Ming and Tauber[13] have pointed out that Sussenguth's algorithm does not take sufficient account of the bond types between particular pairs of atoms (which are considered only as properties of the atoms to which they are attached) and that this can lead to false identification of isomorphisms; they proposed an extension to the algorithm to avoid the problem.

Figueras's Set Reduction Algorithm.[14] In this algorithm, a "characteristic vector" of Boolean values is set up for each query atom, which indicates its possible correspondents in the file structure, based on individual atom properties. The connection table is then used to identify the neighbors of each query atom and the neighbors of their potential correspondents; this allows the creation of a second characteristic vector for each of these neighbors. The two characteristic vectors are then merged by a Boolean AND operation, to "reduce" the sets. If an empty vector results, this indicates that a query

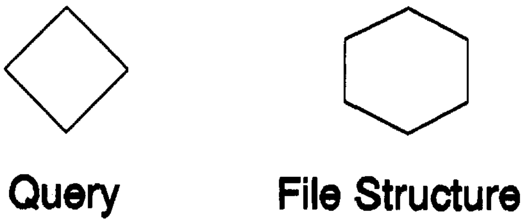


Figure 1. Example of incorrect identification of isomorphisms in the Figueras and von Scholley algorithms. In both the query and file structures, all atoms have identical sets of neighbors; however many atoms distant are examined. The problem would not occur if there were heteroatoms or substituents in the structures.

atom has no possible correspondents and, thus, that there is no subgraph isomorphism. Similarly if the set of file structure correspondents is smaller than the set of query atoms there is no subgraph isomorphism.

The algorithm goes on to use "higher-order" connection tables (listing, for each atom, the atoms n bonds away, for an nth order table) to take account of more distant neighbors and, thus, to further reduce the sets of correspondents. For reasons of ease of generation of the higher-order tables, only tables of order 2^k where $k = 1,2,3$, etc. were used in the original implementation. In contrast to Sussenguth's algorithm, no back-tracking procedure is used when the sets cannot be reduced further, and in certain unusual cases the algorithm may therefore falsely identify isomorphisms; Figure 1 shows an example.

Ullmann's Algorithm.[15] Though it was published in 1976, it was only in the late 1980s that this algorithm began to be used in chemical substructure search applications. Nevertheless, studies by Willett and his students[39,71] have suggested that, for such applications, it may be the most efficient of all subgraph isomorphism algorithms published to date; it is also equally suitable for searching both two- and three-dimensional chemical structure representations.

The algorithm is a combination of a back-tracking procedure with a relaxation-based refinement step and utilizes a Boolean matrix of $N_Q \times N_F$ elements, where N_Q and N_F are the number of nodes in the query substructure and in the file structure, respectively. The elements of the matrix are set to 1 if the specified query node can be mapped to the specified file structure node and to 0 otherwise; the values are initially set on the basis of individual node properties (which may have been obtained from application of some previous algorithm, such as Figueras's). The main part of the algorithm involves (arbitrary) selection of one of the possible correspondents for each query node in turn (i.e., one of the nonzero elements of the matrix row for that query node) and the updating of the matrix by changing all the remaining nonzero elements in that row to zero. If, when trying to select a correspondent for a subsequent query node, it is found that all the nonzero elements in its row have already been provisionally assigned to previous query nodes, the algorithm back-tracks and tries an alternative possible correspondent.

At this level, the algorithm is a simple back-tracking tree search. However, after selection of a correspondent for each query node, a relaxation procedure is applied to refine the matrix and change as many 1's to 0's as possible; if any row is then left with only 0's (i.e., a query node is left with no possible correspondents), the algorithm is able to back up at once, without needing to select further correspondents. The refinement procedure involves inspection of the neighbors of each possible corresponding pair (identified by a 1 in the matrix); expressed informally: if the neighbors of a query node do not correspond (as indicated by 1's in the matrix) with the neighbors of its corresponding file structure node, then the 1 is changed to a 0. This test is repeated iteratively until no further 1's are changed to 0's and, thus, allows rejection of ultimately unproductive mappings at a very early stage.

Von Scholley's Algorithm.[18] The publication of this algorithm in 1984 marked a re-awakening of novel development in the field, after a lull of nearly 10 years. Based on an earlier proposal of Kitchen and Krishnamurthy,[17] in its original implementation it is also capable of handling generic structures, both as query and in the database. The algorithm inverts the initial step of Figueras's algorithm, by building sets of potentially corresponding query structure nodes for each file structure node; each iteration of the algorithm explicitly uses a relaxation technique to reduce the sets of potential correspondents, by examination of the neighbors of each node, and elimination of correspondences if there is an inconsistency between the query and file structure node neighbors. The algorithm is iterated until either one of the query nodes disappears from all the correspondence sets (in which case there is no isomorphism) or until no further eliminations are made (in which case there is very likely to be an isomorphism). Like Figueras's algorithm, and in contrast to those of Sussenguth and Ullmann, von Scholley's algorithm does not include a back-tracking procedure as a fall-back; it may therefore falsely identify some isomorphisms, such as that in Figure 1.

SCREENING

Although the algorithms described in the preceding section offer considerable improvements over brute force or basic back-tracking algorithms, they remain very time-consuming and, thus, impractical for searching more than a few thousand compounds on conventional computers. Screening systems allow the bulk of the compounds in a database, which cannot match the query, to be eliminated rapidly from consideration, so that only a restricted number of candidates need to be examined more rigorously.

Screening systems normally involve indexing the compounds in a database by use of a set of search keys, each of which describes some structural feature of the molecule. A corresponding set of keys can be identified for the query structure, and those file structures which do not contain the keys identified for the query can be eliminated from further consideration.

The choice of keys is, of course, crucial. It is a general principle of indexing systems that middle-frequency keys are the most useful: very commonly occurring ones are of no use because they do not allow any structures to be eliminated; very uncommon ones are of no use because it is unlikely that they will ever occur in queries. The keys must also, as far as possible, be independent of each other, since if one key always or nearly always occurs with another, it contributes nothing to the retrieval performance.

In the early 1970s, Lynch and his colleagues studied a range of substructural fragment types and their statistical occurrence in large files of compounds;[24–26] this approach was used for the establishment of a fragment dictionary by the BASIC group of companies in Switzerland,[27] a dictionary which was later adopted by STN International for its online substructure search system.[28] Work has also been done on automatic selection of fragments on the basis of their statistical occurrence in the file to be searched.[29–32]

The normal approach to matching query and file structures for screening, where a fixed dictionary of search keys is being used, is to compare bit strings in which each bit represents one key; greater search speed can be obtained by using an inverted bit map, though where the set of search keys is open-ended, this approach is not feasible. Another approach to fragment screen searching, using hierarchically structured inverted files, is discussed later.

The storage requirements for the bit string (and thus the search time) can be reduced, especially where a large number of keys are being used, by using the principle of "superimposition".[30,33] In this, a long bit string is "folded" to form a shorter one, each bit position in the longer bit string giving rise to several bits in the shorter one. While this can lead to failure to exclude compounds which do not contain the required search keys (because different keys can cause the same bits to be set in the short bit string), this happens only infrequently provided that only about half the bits in the short bit string are set.

HARDWARE SOLUTIONS

The modern availability of computers with very large amounts of random-access memory has provided an opportunity for further application of the superimposition principle in the substructure search system marketed by Daylight Chemical Information Systems Inc.[34] The reduction in the length of bit string (or "fingerprint") required for each compound enables the entire bit map, even for very large databases, to be held in memory and, thus, permits very rapid screening searches.

A prototype memory-based substructure search system is also under development at Chemical Abstracts Service,[35] though other hardware solutions to the substructure search problem have generally used parallel processing approaches and have been under active investigation since the mid-1980s when suitable machines started to become available.

At the simplest level, these involve dividing the database to be searched into several portions, each to be searched on a different machine, generally using the normal algorithms discussed earlier. The best-known example of this is the use of pairs of minicomputers in parallel for the substructure searching of the Chemical Abstracts Registry File on the STN International System;[36] though overall control is needed to collate the results from each individual machine, this essentially involves only the division of a large file into manageable subsets, which can be searched in acceptable time.

A more sophisticated example of this "database-parallel" approach has been described by Jochum and Worbs, in an experimental system called TOPFIT.[37] In a specially designed hardware configuration of 8-bit processors, a "master" processor feeds connection tables to a series of "slave" processors, which carry out the matching process using a conventional algorithm. Similar investigations have been carried out at Sheffield University using networks of Inmos Transputer processors (in various configurations);[38,39] the algorithms used were those of Sussenguth, Figueras, and von Scholley. In all this work, the ratio of speedup to the number of processors falls off as the number of processors is increased and the overheads involved in controlling them and in transmitting the database connection tables to them become greater; this does, however, only reflect the relative processor and communication speeds in the hardware being used.

An alternative approach is algorithmic parallelism, in which the operations of the algorithm itself are spread over multiple processors and carried out simultaneously, thus speeding up the search of each individual database structure. Conventional computers have been used to simulate the parallel implementation of back-tracking[40] and of the von Scholley algorithm,[41] and the latter has also been implemented in an algorithmically parallel version on a transputer network.[39] The von Scholley algorithm is particularly well-suited to parallel implementation and indeed was originally developed with parallelism in mind: each iteration of the relaxation step, in which the neighbors of each atoms are examined, can be carried out in parallel for all atoms. The main problem which was found is that the actual operations which can be carried out in parallel are relatively simple, and thus given the relatively slow communication speeds available in transputer networks, the processors spend much of their time awaiting the next data.

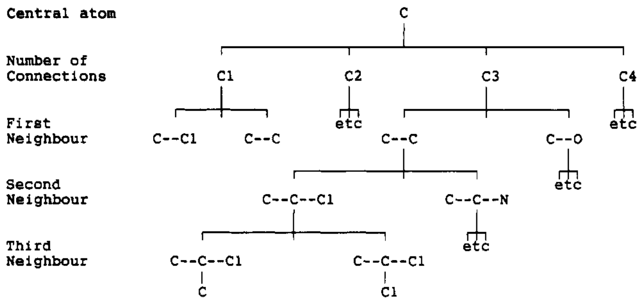


Figure 2. Hierarchical fragment descriptions used in the CIS (Feldmann) substructure search system. Each level in the hierarchy enlarges the description of the fragment.

The relaxation-based refinement step in the Ullmann algorithm is also particularly appropriate for parallel implementation, and this has been investigated by Willett et al.[42] using the AMT Distributed Array Processor (DAP). Unlike the parallel hardware configurations used in other work, the DAP is a so-called *single instruction multiple data* (SIMD) machine, in which the same instructions are executed simultaneously on different data by a large number of very simple processors (4096 in the machine used in this case). This architecture allows the Boolean matrix used by the Ullmann algorithm to be stored one element per processor and the refinement step to be applied to each element simultaneously. Though the experiments reported[42] found the algorithm-parallel implementation to be faster than a database-parallel implementation on the same hardware, it was suggested that a mixed approach might be the most successful.

TREE-STRUCTURED FRAGMENT SEARCHES

Two systems developed in the 1970s employ an hierarchical tree structure for substructural fragment-based screening. In the Chemical Information System (CIS) developed at the National Institutes of Health,[43] there are separate searches for atom-centered fragments and ring system descriptors. The fragment search proceeds from the central atom type via the number of neighbors it has to the atom type and bond orders for each neighbor in turn (see Figure 2); in the ring systems search, successive levels of the search tree describe the ring pattern, the atom types present, the heteroatom positions, and the ring substitution positions. An atom-by-atom back-tracking search is available for rigorous searching of those compounds retrieved by the fragment and ring probe searches.

In the DARC system,[44] the fragments (called FRELs, Fragment Reduced to an Environment which is Limited) describe two concentric "layers" of atoms around a *focus*, which is an atom with at least three (or in some cases two) non-hydrogen neighbors. The FRELs generated from the structures in a database are also stored in an hierarchical tree, with generalized forms called *fuzzy FRELs* being included at higher levels of the tree. In searching, the FREL search (which uses only as many FRELs from the query as is necessary to reduce the number of candidate database structures to an acceptable number) is followed by a bit screen match, mainly concerned with ring systems, and a back-tracking atom-by-atom match.

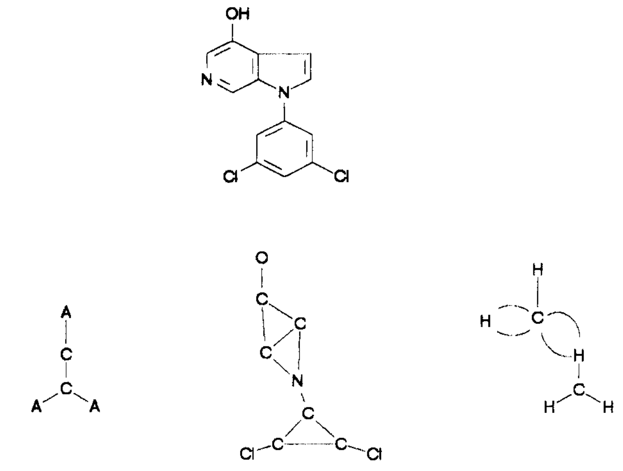
The main recent developments of new algorithms for substructure searching have also employed searches in tree-structured fragment files. The Hierarchical Tree Substructure Search (HTSS) system was developed in Hungary[45,46] and was used for substructure search of the Beilstein database on the Orbit online system, though this service was withdrawn in Sept 1992. Each level in the hierarchical fragment tree is effectively part of an hierarchical classification of all the atoms in the database as a whole, initially by number of neighbors and atom type, and then by bonding pattern and atom type of neighbors. The size of any rings in which the atoms occur is also taken into account, and lower levels of the tree are generated by applying a relaxation procedure, subdividing the atoms on the basis of the classifications already applied to their neighbors, and thus taking account of successively more distant atoms. This concentrates the computationally intensive work into creation of the hierarchical search files, leading to rapid searches, and effectively obviating the need for an atom-by-atom search stage.

The S4 system was developed by Softron GmbH, in association with the Beilstein Institute, and is used for substructure search of the Beilstein database on the Dialog online system[47] and in the Beilstein current Facts CD-ROM product. In the S4 system,[48-50] very compact codes are generated (again using a relaxation procedure) for each atom in the database and are sorted into a hierarchical search tree. The tree is arranged so that the minimum number of disk accesses is required during search, which further increases the speed, especially in CD-ROM implementations. A very similar approach has been used in the ReSy system, developed internally for in-house use by Bayer AG at Leverkusen, FRG.

The development of systems of this type during the mid-1980s has been a major advance in substructure searching software and is now allowing large databases to be searched in acceptable time on desktop microcomputers. The superior performance of this type of tree-structured search, in which much of the computational effort is put into pre-processing the database, over the conventional fragment screen and back-tracking algorithm approach has been indicated by a comparison of systems carried out at the Beilstein Institute.[48,49] However, the practical problems in adequately testing all the systems under identical conditions make it difficult to reach firm conclusions about the "best" system in all situations; certainly it seems that the performance of some systems can vary considerably with the nature of the substructure queries submitted.

OTHER WORK

Two other approaches to two-dimensional substructure searching have been examined in recent years: the use of *reduced graph* representations of individual structures and the creation of *hyperstructures* encompassing all the structures in a database. In addition, there has been a rapidly growing interest in the searching of databases of three-dimensional structure representations.[51]



Cyclic/Acyclic Homeomorphic Carbon/Heteroatom

Figure 3. Structure with three different types of reduced graphs which may be derived from it. Note the two bonds joining some pairs of nodes in the carbon/heteroatom reduced graph.

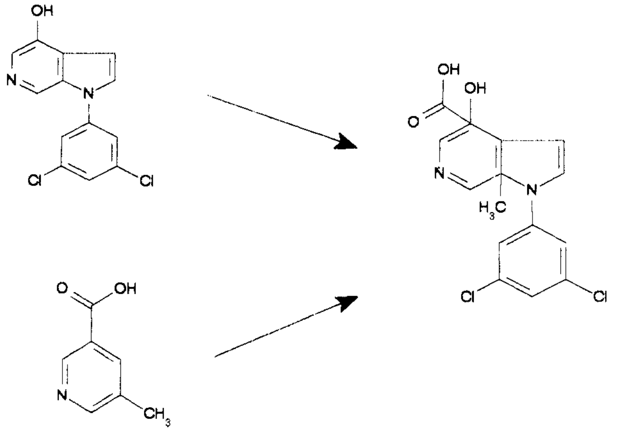


Figure 4. Two normal structures, and a hyperstructure which may be formed from them, in which their common parts are shown only once. Note the 5-valent carbon atoms which occur as a result of different substitution patterns on the common ring.

Reduced Graphs. In a reduced graph, groups of atoms within the structure are collapsed together to form single nodes, and the smaller graphs which result can be searched more quickly, using any of the conventional methods. The reduced graph may contain nodes representing the cyclic and acyclic portions of the molecule or contiguous groups of carbon or heteroatoms; another type of reduced graph which has been studied[52] is the homeomorphically reduced graph formed by deleting all atoms of connectivity 2. Figure 3 shows examples of various types of reduced graph. For normal substructure searching of databases of specific molecules, reduced graph searches, because of the loss of information inherent in the formation of the reduced graph, can at best operate only as screening searches and, for the relatively simple query substructures normally encountered, are both computationally more expensive and less effective than conventional fragment-based screening.[52] Reduced graphs have, however, found useful application in searching of generic structures;[53,54] recent developments in systems for searching databases of generic structures have been reviewed by me[55] and are not discussed here.

Hyperstructures. This concept was introduced by Vladutz and Gould[56] in 1987 and suggested as a means for both compact storage and rapid searching of large files of chemical structures. A hyperstructure is formed by superimposing several molecules on each other, so that those parts which are common to them need be stored only once. This can be expected not only to reduce storage requirements, but it also allows more rapid searching. Figure 4 shows an example. This is because a single subgraph isomorphism with the hyperstructure may identify a large number of hits, though *all* the isomorphic mappings of the substructure to the hyperstructure must be found to identify all the hits. Methods for building and searching hyperstructures are currently under investigation at Sheffield University.[57] Two generation methods have been attempted, though one has proved computationally infeasible for more than a few tens of structures. The other produces a different and more highly connected hyperstructure, which may be less efficient for searching, and substructure searches using the Ullmann algorithm have given mixed results. Very recent work[58] has used a novel genetic algorithm to generate hyperstructures which appears more encouraging.

Three-Dimensional Structure Searching. This is of particular importance in drug design, as searches can be made for the actual three-dimensional pattern of atoms and bonds believed to interact with a biological receptor site and thus to exert some pharmacological effect.[59,60] A number of approaches have been studied for searching for such pharmacophoric patterns or *pharmacophores*.[61]

Searching based on the specification of bond length and angle ranges has been developed for use with the Cambridge Crystal Structure Database;[62,63] another approach is based on analyzing the shape of the pharmacophore in terms of overlapping spheres.[64] More recent work has used an approach analogous to the substructure searching of files of two-dimensional structure representations; in this case a topological graph can be constructed for the three-dimensional structure in which all the nodes (atoms) are connected to each other, with the edges representing the interatomic distances. Pharmacophoric pattern matching, thus, becomes a process of subgraph isomorphism within this fully connected graph.

Lesk has described an algorithm[65] in which structure atoms which cannot match pattern (query) atoms are iteratively removed from consideration, after which the brute-force approach of examining all possible combinations of structure and pattern atoms is used. This is clearly very expensive computationally and, except for very simple cases, is not feasible.

A second procedure which has been tried is based on the detection of *cliques* (subgraphs where every node is connected to every other node) in the so-called *correspondence* graph, which is formed from the pattern and structure graphs by using each pair of nodes (one from each) as nodes. Edges in the correspondence graph occur where the edges between the nodes in the original pattern and structure graphs are the same (i.e., represent the same interatomic distance).[66,67]

In addition to these methods, the algorithms used for subgraph isomorphism in two-dimensional structure representations have also been used;[68–70] a comparison of algorithms by Brint and Willett[71] clearly showed the superiority of the Ullmann algorithm. Work has also been done on the use of three-dimensional screens to limit the number of structures needing to be processed by a full geometric searching algorithm,[32,72–74] and this approach has been incorporated into a number of recently developed operational systems.[75–79]

SUMMARY AND CONCLUSIONS

This review has described the techniques which have been developed over the past 3.5 decades to carry out substructure searches in representations of chemical structures. This is an area which saw considerable activity in the 1960s and early 1970s, as the main algorithms were identified; it was followed by a period of consolidation during which they were incorporated into operational systems, based on interactive graphical input of queries and output of search results. Since the mid-1980s, the area has again become an active one for research, with significant new advances being made for both two- and three-dimensional searching. It is likely that the remaining years of the century will see the appearance of new operational systems in which very large chemical structure databases, including both two- and three-dimensional representations, can be searched on desktop computers.

REFERENCES AND NOTES

(1) Tarjan, R. E. Graph algorithms in chemical computation. In *Algorithms for chemical computations*; Christoffersen, R. E., Ed.; *ACS Symposium Series 46*; American Chemical Society: Washington, DC, 1977; pp 1–19.

(2) Willett, P. A review of chemical structure retrieval systems. *J. Chemom.* **1987**, *1*, 139–155.

(3) Barnard, J. M. Problems of substructure search and their solution. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 113–126.

(4) Read, R. C.; Corneil, D. G. The graph isomorphism disease. *J. Graph Theor.* **1977**, *1*, 339–363.

(5) Barnard, J. M. Structure representation and searching. In *Chemical Structure Systems*; Ash, J. E., Warr, W. A., Willett, P., Eds.; Ellis Horwood: Chichester, 1991.

(6) Cook, S. A. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*; ACM: 1971; pp 151–158.

(7) Karp, R. M. On the computational complexity of combinatorial problems. *Networks* **1975**, *5*, 45–68.

(8) Gati, G. Further annotated bibliography on the isomorphism disease. *J. Graph Theor.* **1979**, *3*, 95–109.

(9) Ray, L. C.; Kirsch, R. A. Finding chemical records by digital computers. *Science* **1957**, *126*, 814–819.

(10) Dengler, A.; Ugi, I. A central atom based algorithm and computer program for substructure search. *Comput. Chem.* **1991**, *15*, 103–107.

(11) Jun, X.; Maosen, Z. HBA: new algorithm for structural match and applications. *Tetrahedron Comput. Methodol.* **1989**, *2*, 75–83.

(12) Sussenguth, E. H. A graph-theoretic algorithm for matching chemical structures. *J. Chem. Doc.* **1965**, *5*, 36–43.

(13) Ming, T. K.; Tauber, S. J. Chemical structure and substructure search by set reduction. *J. Chem. Doc.* **1971**, *11*, 47–51.

(14) Figueras, J. Substructure search by set reduction. *J. Chem. Doc.* **1972**, *12*, 237–244.

(15) Ullmann, J. R. An Algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31–42.

(16) McGregor, J. J. Relational consistency algorithms and their application to finding subgraph and graph isomorphisms. *Inf. Sci.* **1979**, *19*, 229–250.

(17) Kitchen, L.; Krishnamurthy, E. V. Fast, parallel relaxation screening for chemical patent database search. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 44–48.

(18) Von Scholley, A. A relaxation algorihm for generic chemical structure screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 235–241.

(19) Morgan, H. L. The generation of a unique machine description for chemical structures—a technique developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–113.

(20) Liu, X.; Balasubramanian, K.; Munk, M. E. Computational techniques for vertex partitioning of graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 263–269.

(21) Rücker, G.; Rücker, C. On using the adjacency matrix power method for perception of topological symmetry and for isomorphism testing of highly intricate graphs. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 123–126.

(22) Figueras, J. Automorphism and equivalence classes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 153–157.

(23) Unger, S. H. GIT—a heuristic program for testing pairs of directed line graphs for isomorphism. *Commun. Assoc. Comput. Mach.* **1964**, *7*, 26–34.

(24) Lynch, M. F. Screening large chemical files. In *Chemical information systems*; Ash, J. E., Hyde, E., Eds.; Ellis Horwood: Chichester, 1974; pp 177–194.

(25) Lynch, M. F. The microstructure of chemical databases and the choice of representation for retrieval. In *Computer representation and manipulation of chemical information*; Wipke, W. T., Heller, S. R., Feldmann, R. J., Hyde, E., Eds.; Wiley: New York, 1974; pp 31–53.

(26) Ash, J. E.; Chubb, P. A.; Ward, S. E.; Welford, S. M.; Willett, P. *Communication, storage and retrieval of chemical information*; Ellis Horwood: Chichester, 1985; pp 160–167.

(27) Graf, W.; Kaindl, H. K.; Kniess, H.; Warszawski, R. The third BASIC fragment search dictionary. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 177–181.

(28) Dittmar, P. G.; Farmer, N. A.; Fisanick, W.; Haines, R. C.; Mockus, J. The CAS Online search system. 1. General system design and selection, generation and use of search screens. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 93–102.

(29) Feldman, A.; Hodes, L. An efficient design for chemical structure searching. 1. The screens. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147–152.

(30) Hodes, L. Selection of descriptors according to discrimination and redundancy. Application to chemical structure searching. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 88–93.

(31) Willett, P. A screen set generation algorithm. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 159–162.

(32) Cringean, J. K.; Pepperrell, C. A.; Poirrette, A. R.; Willett, P. Selection of screens for three-dimensional substructure searching. *Tetrahedron Comput. Methodol.* **1990**, *3*, 37–46.

(33) Meyer, E. Superimposed screens for the GREMAS system. In *Mechanised information storage retrieval and dissemination*; Proceedings of the FID-IFP Conference, Rome, June 14–17, 1967; Samuelson, K., Ed.; North Holland: Amsterdam, 1968; pp 280–288.

(34) Daylight Chemical Information Systems, Inc., 18500 Von Karman Avenue, Suite 450, Irvine, CA 92715.

(35) Lillie, D. H.; Rusinko, A. A memory-based structure search system prototype. Presented at the 204th ACS National Meeting, Washington, DC, Aug 1992. Submitted to *J. Chem. Inf. Comput. Sci.*

(36) Farmer, N.; Amoss, J.; Farel, W.; Fehribach, J.; Zeidner, C. R. The evolution of the CAS parallel structure searching architecture. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 283–296.

(37) Jochum, P.; Worbs, T. A multiprocessor architecture for substructure search. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 279–282.

(38) Brint, A. T.; Gillet, V. J.; Lynch, M. F.; Willett, P.; Manson, G. A.; Wilson, G. A. Chemical graph matching using transputer networks. *Parallel Comput.* **1988**, *8*, 295–300.

(39) Downs, G. M.; Lynch, M. F.; Willett, P.; Manson, G. A.; Wilson, G. A. Transputer implementations of chemical substructure searching algorithms. *Tetrahedron Comput. Methodol.* **1988**, *1*, 207–217.

(40) Wipke, W. T.; Rogers, D. Rapid subgraph search using parallelism. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 255–262.

(41) Gillet, V. J.; Welford, S. M.; Lynch, M. F.; Willett, P.; Barnard, J. M.; Downs, G. M.; Manson, G.; Thompson, J. Computer storage and retrieval of generic chemical structures in patents. 7. Parallel simulation of a relaxation algorithm for chemical substructure search. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 118–126.

(42) Willett, P.; Wilson, T.; Reddaway, S. F. Atom-by-atom searching using massive parallelism. Implementation of the Ullmann subgraph isomorphism algorithm on the Distributed Array Processor. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 225–233.

(43) Feldmann, R. J.; Milne, G. W. A.; Heller, S. R.; Fein, A.; Miller, J. A.; Koch, B. An interactive substructure search system. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 157–163.

(44) Attias, R. DARC substructure search system: a new approach to chemical information. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 102–108.

(45) Bruck, P.; Nagy, M. Z.; Kozics, S. Substructure search on hierarchical trees. In *Online information 87*; Proceedings of the 11th international online information meeting, London, Dec 8–10, 1987; Learned Information: Oxford, 1987; pp 41–43.

(46) Nagy, M. Z.; Kozics, S.; Veszpremi, T.; Bruck, P. Substructure search on very large files using tree-structured databases. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 127–130.

(47) Hartwell, I. O.; Haglund, K. A. An overview of Dialog. In *The Beilstein Online Database: implementation, content and retrieval*; Heller, S. R., Ed.; ACS Symposium Series 436; American Chemical Society: Washington, DC, 1990; pp 42–63.

(48) Hicks, M. G.; Jochum, C. Substructure search systems. 1. Performance comparison of the MACCS, DARC, HTSS, CAS Registry, MVSSS, and S4 substructure search systems. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 191–199.

(49) Hicks, M. G.; Jochum, C.; Maier, H. Substructure search systems for large chemical data bases. *Anal. Chim. Acta* **1990**, *235*, 87–92.

(50) Bartmann, A.; Maier, H.; Roth, B.; Walkowiak, D. Substructure search on very large files by using multiple storage techniques. Presented at the 204th ACS National Meeting, Washington, DC, Aug 1992. Submitted to *J. Chem. Inf. Comput. Sci.*

(51) Martin, Y. C.; Bures, M. G.; Willett, P. Searching databases of three-dimensional structures. In *Reviews in computational chemistry*; Lipkowitz, K. B., Boyd, D. B., Eds.; VCH: New York, 1990; pp 213–263.

(52) Cringean, J. K.; Lynch, M. F. Subgraphs of reduced chemical graphs as screens for substructure searching of specific chemical structures. *J. Inf. Sci.* **1989**, *15*, 211–222.

(53) Gillet, V. J.; Downs, G. M.; Ling, A. B.; Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 8. Reduced chemical graphs, and their application in generic chemical structure retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126–137.

(54) Fisanick, W. The Chemical Abstracts Service generic chemical (Markush) structure storage and retrieval capability. I. Basic Concepts. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 145–154.

(55) Barnard, J. M. A comparison of different approaches to Markush structure handling. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 64–68.

(56) Vladutz, G.; Gould, S. R. Joint compound/reaction storage and retrieval and possibilities of a hyperstructure-based solution. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 371–383.

(57) Brown, R. D.; Downs, G. M.; Willett, P.; Cook, A. P. F. A hyperstructure model for chemical structure handling: generation and atom-by-atom searching of hyperstructures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 522–531.

(58) Brown, R. D. Personal communication, 1992.

(59) Gund, P. Three-dimensional pharmacophoric pattern searching. *Prog. Mol. Subcell. Biol.* **1977**, *5*, 117–143.

(60) Esaki, T. Quantitative drug design studies. V. Approach to lead generation by pharmacophoric pattern searching. *Chem. Pharm. Bull.* **1982**, *30*, 3657–3661.

(61) Willett, P. *Three-dimensional chemical structure handling*; Wiley: New York, 1991.

(62) Murray-Rust, P.; Motherwell, S. Computer retrieval and analysis of molecular geometry. 1. General principles and methods. *Acta Crystallogr.* **1978**, *B34*, 2518–2526.

(63) Allen, F. H. The Cambridge structural database as a research tool in chemistry. In *Modelling of structure and properties of molecules*; Maksic, Z. B., Ed.; Ellis Horwood: Chichester, 1987; pp 51–66.

(64) Sheridan, P.; Venkataragharvan, R. Designing novel nicotinic agonists by searching a database of molecular shapes. *J. Comput.-Aided Mol. Des.* **1987**, *1*, 243–256.

(65) Lesk, A. M. Detection of three-dimensional patterns of atoms in chemical structures. *Commun. Assoc. Comput. Mach.* **1979**, *22*, 219–224.

(66) Golender, V.; Rosenblit, A., Eds. *Logical and combinatorial algorithms for drug design*; Research Studies Press: Letchworth, 1983.

(67) Kuhl, F. S.; Crippen, G. M.; Friesen, D. K. A combinatorial algorithm for calculating ligand binding. *J. Comput. Chem.* **1984**, *5*, 24–34.

(68) Jakes, S. E.; Watts, N.; Willett, P.; Bawden, D.; Fisher, J. D. Pharmacophoric pattern matching in files of three-dimensional chemical structures. Evaluation of search performance. *J. Mol. Graphics* **1987**, *5*, 41–48.

(69) Brint, A. T.; Mitchell, E.; Willett, P. Substructure searching in files of three-dimensional chemical structures. In *Chemical Structures: The International Language of Chemistry*; Proceedings of an International Conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31–June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg 1988; pp 131–144.

(70) Grindley, H. M.; Mitchell, E. M.; Willett, P.; Artymiuk, P. J.; Rice, D. W. Graph-matching techniques for databases of three-dimensional structures. In *Chemical information systems beyond the structure diagram*; Bawden, D., Mitchell, E. M., Eds.; Ellis Horwood: Chichester, 1990; pp 50–62.

(71) Brint, A. T.; Willett, P. Pharmacophoric pattern matching in files of 3D chemical structures: comparison of geometric searching algorithms. *J. Mol. Graphics* **1987**, *5*, 49–56.

(72) Jakes, S. E.; Willett, P. Pharmacophoric pattern matching in files of three-dimensional chemical structures. Selection of interatomic distance screens. *J. Mol. Graphics* **1986**, *4*, 12–20.

(73) Clark, D. E.; Willett, P.; Kenny, P. W. Pharmacophoric pattern matching in files of three-dimensional chemical structures: use of smoothed bounded distances for incompletely-specified query patterns. *J. Mol. Graphics* **1991**, *9*, 157–160.

(74) Poirrette, A. R.; Willett, P.; Allen, F. H. Pharmacophoric pattern matching in files of three-dimensional chemical structures: characterisation and use of generalised valence angle screens. *J. Mol. Graphics* **1991**, *9*, 203–217.

(75) Van Drie, J. H.; Weininger, D.; Martin, Y. C. ALADDIN: an integrated tool for computer-assisted molecular design and pharmacophore recognition from geometric, steric and substructure searching of three-dimensional molecular structures. *J. Comput.-Aided Mol. Des.* **1989**, *3*, 225–251.

(76) Sheridan, R. P.; Rusinko, A.; Nilakantan, R.; Venkataraghavan, R. Searching for pharmacophores in large co-ordinate databases and its use in drug design. *Proc. Natl. Acad. Sci. U.S.A.* **1989**, *86*, 8165–8169.

(77) Sheridan, R. P.; Nilakantan, R.; Rusinko, A.; Bauman, N.; Haraki, K. S.; Venkataragharvan, R. 3DSEARCH, a system for three-dimensional substructure searching. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 255–260.

(78) Moock, T. E.; Christie, B.; Henry, D. MACCS-3D: a new database system for three-dimensional molecular models. In *Chemical information systems beyond the structure diagram*; Bawden, D., Mitchell, E. M., Eds.; Ellis Horwood: Chichester, 1990; pp 42–49.

(79) Güner, O. F.; Hughes, D. W.; Dumont, L. M. An integrated approach to three-dimensional information management with MACCS-3D. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 408–414.