Journal of
Applied Crystallography

ISSN 0021-8898

Received 13 January 2004
Accepted 5 February 2004

A new algorithm for performing three-dimensional searches of the Cambridge Structural Database

James A. Chisholm[a,b]* and Sam Motherwell[a]

[a]Cambridge Crystallographic Data Centre, 12 Union Road, Cambridge CB2 1EZ, UK, and [b]Pfizer Institute for Pharmaceutical Materials Science, University of Cambridge, Department of Materials Science and Metallurgy, Pembroke Street, Cambridge CB2 3QZ, UK. Correspondence e-mail: chisholm@ccdc.cam.ac.uk

A search algorithm, 3*DSEARCH*, is presented that can readily identify challenging extended chemical queries from three-dimensional molecular crystal structure information. The program combines substructure search and distance search techniques within a depth-first backtracking algorithm. Performance metrics are presented for example searches composed of several substructures and several intermolecular connections. It is shown that such searches, which are outside the capabilities of current search engines, can now be performed on the entire Cambridge Structural Database with search times of around half an hour.

1. Introduction

The Cambridge Structural Database (CSD; Allen, 2002) contains chemical and structural information for over 300 000 organic and metal-organic crystal structures. Such data, together with associated search, visualization and analysis software, has long been considered a valuable research tool in fields such as crystallography, crystal engineering, drug design and molecular modelling. A common task is to search the CSD for a chemical query that is subject to various geometric constraints, such as constraints on bond lengths, bond angles, torsion angles and interatomic distances. Such searches can be performed with the search program *ConQuest* (Bruno *et al.*, 2002). *ConQuest* performs exceptionally well and can readily identify substructures and queries containing first-neighbour molecular contacts.

However, the task of searching the three-dimensional coordinates can become problematic for large search queries that span several molecules. For example, the task of identifying extended hydrogen-bond ring motifs can take several hours or even days with *ConQuest*, and the search engine will not identify all occurrences of the motif. We present here a new search algorithm, 3*DSEARCH*, that can exhaustively search three-dimensional crystal coordinates for challenging patterns in an efficient and accurate manner. Performance metrics are presented for two searches that demonstrate the ability of 3*DSEARCH* to identify queries containing several intermolecular contacts.

2. Methodology

Fig. 1 shows the generalization of a query. A query can be viewed as a collection of *substructures* and *connections* plus a list of constraints, such as distance, angle and torsion constraints. Substructures are covalently bonded fragments defined by atom and bond types. Connections specify a distance criterion between two atoms, such as a distance less than the sum of the van der Waals radii for the two atoms (a close contact). Finding a query within a crystal structure involves finding the constituent components, *viz.* the substructures and the connections.

2.1. Substructure and connection searches

To identify individual substructures we employ the Ullmann (1976) algorithm. This is an efficient subgraph isomorphism algorithm, and as both query substructure and target molecule can be represented as graphs, this algorithm allows us to identify all occurrences of a substructure within a given molecule. This graph-matching algorithm attempts to match query and target nodes by comparing node connectivity. In our case a modified Ullmann algorithm is employed, which, in addition to node connectivity, checks that chemical information, such as atom types, bond types and charge constraints, also matches between query and target.

To find individual connections, we implement a procedure based on a distance search algorithm described by Rollett (1965). The task is, given a unit cell, space group and asymmetric unit, to find all atomic positions that are within a specified distance from a reference atomic position. To begin, a filled unit cell of coordinates is constructed by application of the appropriate symmetry operations on the asymmetric unit. Then the dimensions of the unit cell and the distance criterion are used to determine necessary maximum and minimum unit-cell translations. These translations are applied to all relevant atomic positions in the reference unit cell and the resulting test points are checked against the distance criterion. This distance search procedure is quite general and can identify arbitrarily large distances as well as close contacts and intramolecular distances.

2.2. The backtracking algorithm

The 3*DSEARCH* algorithm combines these substructure and connection search techniques into a depth-first backtracking algorithm to identify whether all substructures and all connections in the query can be found in a crystal structure. The algorithm begins by searching whole molecules belonging to the asymmetric unit to determine whether the first substructure in the query can be found (there may be several matches). The algorithm then decides which substructure to search for next. The only condition used is that the next substructure must form at least one connection to a substructure that has already been found. In this way the algorithm works outwards from an initial starting point to build up a query match. The

search for a subsequent substructure is given the name 'find next' and is composed of three main steps:

1. Find connections.
2. Find substructures.
3. Find remaining connections.

These steps are shown schematically in Fig. 2. Both step 1 and step 2 may produce a list of hits, and all hits must be visited to ensure that the search is exhaustive. In step 3, the term 'remaining connections' refers to connections that connect to the current substructure, have not yet been found and connect to a substructure that has been found. This step is a very quick and simple check rather than a search procedure. Thus the presence of many connections in a query does not necessarily imply the need for many costly connection searches.

Constraints, such as angle constraints and distance constraints, are checked as soon as it is possible to do so. If at any point a search step fails, the algorithm backtracks, as indicated by the double headed arrows in Fig. 2, and the next avenue is searched. Individual substructure hits and connection hits can be represented as nodes in a search tree, and a particular combination of hits, such as the combination shown in Fig. 2, can be represented by a particular branch. The search algorithm stops once all branches in the search tree have been traversed or once the specified number of query matches has been found.

It is worth noting that the 'remaining connections' in step 3 can be thought of as constraints. The checking of these constraints leads to a reduction of the search space. Thus the presence of many connections can significantly reduce search times.

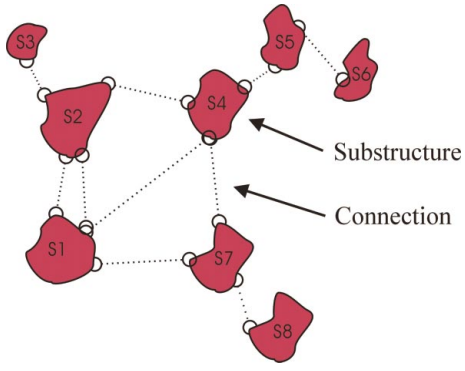

Figure 1
A generalized query viewed as a collection of substructures and connections. Queries can be subject to a list of constraints, such as constraints on atom and bond types, constraints on connection types (inter/intramolecular), and distance, angle and torsion constraints.

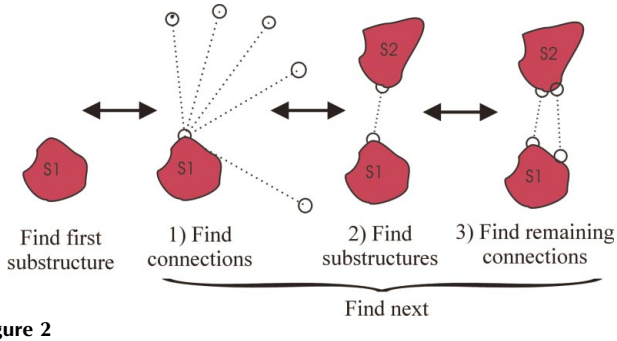

Figure 2
The process of building up a query match by repetition of the step 'find next'. The algorithm backtracks when a search step fails, as indicated by the double-headed arrows.

2.3. Efficiency issues

Two points related to efficiency are worth noting. Firstly, the atom types at the end of each connection are used to speed subsequent substructure searches, as the type provides a useful known node match in the graph-matching algorithm that can be used to reduce the size of the initial Ullmann matrix. This known, or necessary, node match is especially beneficial when searching large molecules.

One potential difficulty with a backtracking approach is that the search tree can quickly become large as the number of substructures in the query increases and as the number of query components present in the structure increases. Our approach for tackling large search trees is to ensure that the traversal of individual nodes is as efficient as possible. One effective way to achieve efficiency is to reuse search results. When the algorithm backtracks, information gained about the crystal is not thrown away; rather the results of substructure and connection searches are stored for later use. This procedure is analogous to dropping breadcrumbs when exploring a maze. Before embarking on a search, breadcrumbs can be checked to determine whether the algorithm has performed the search before. In this way the same searches are never repeated and the traversal of individual nodes in the search tree becomes extremely efficient.

This point is demonstrated in Fig. 3, which shows the top half only of a search tree traversed while searching for a four-membered ring of water molecules. The tree has a depth of seven, which corresponds to the four substructure and three connection searches that are required in order to identify the motif. The dark nodes represent potentially lengthy connection and substructure searches, and the light-grey nodes represent the retrieval of previous search results. Fig. 3 shows how the relatively costly dark searches are performed early on in the

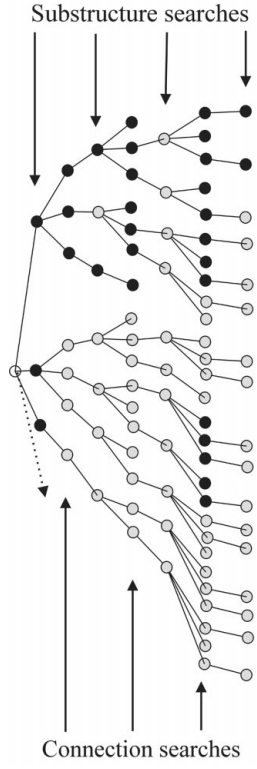

Figure 3
The top section of the search tree traversed while searching for a four-membered ring of water molecules. The dark nodes represent connection and substructure searches. The light-grey nodes represent instances when stored search results have been reused. The dotted arrow indicates that the search tree has a lower section. The nodes in this lower section are all light grey.

search tree and how these can turn quickly into the more efficient light-grey steps. The bottom half of this search tree (not shown) is all light grey. Fig. 3 also shows that branches have been cut. It is important that branches be cut as soon as it is possible to do so by checking that partial query matches satisfy all relevant constraint conditions.

3. Performance

To demonstrate performance we present search times for two example searches, both of which contain several substructures and span several molecules. All searches were performed on the CSD (272 065 entries) using a PC with a 1.8 GHz Pentium 4 processor. The following filters were used: three-dimensional coordinates were determined, and no disorder, no errors and no polymeric entries were allowed.

We note that strategies for searching databases typically employ efficient pre-screen steps, which quickly rule out structures as possible matches. In this way the need to even begin a three-dimensional search for a large number of structures can normally be avoided. The development of such pre-screen steps has not been the focus of this work, and as such, the timing information presented reflects the efficiency of the 3*DSEARCH* algorithm alone.

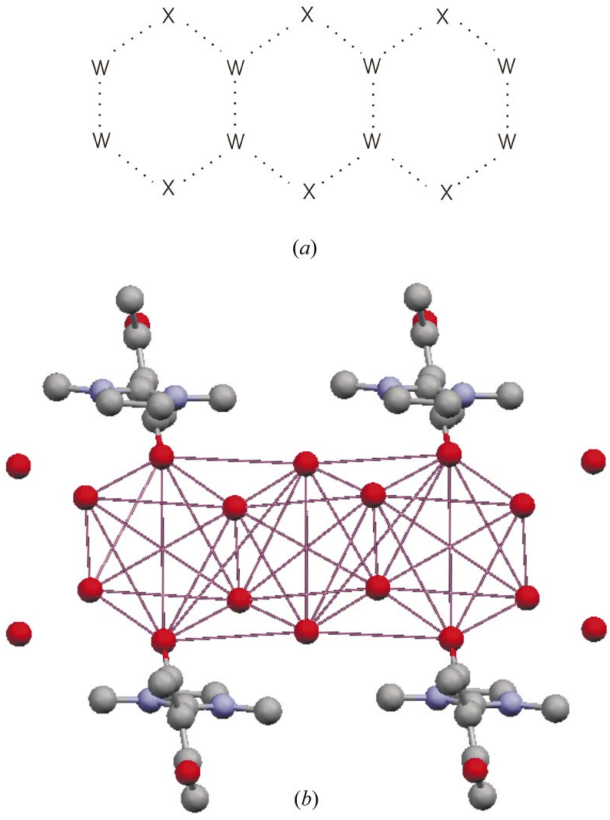

Figure 4
(*a*) A tape motif composed of a specific arrangement of water molecules (marked by letter W) and general atoms (marked by letter X). General atoms can match any element except C and H. Dotted lines indicate hydrogen bonds. This motif falls into the category *T*6(2). (*b*) The CSD entry AMIMZC10, showing the presence of the *T*6(2) motif. The dark-purple lines represent query connections. Two types of connections are specified: short connections with distance criteria less than the sum of the van der Waals radii and long connections greater than the sum of the van der Waals radii plus 0.35 Å.

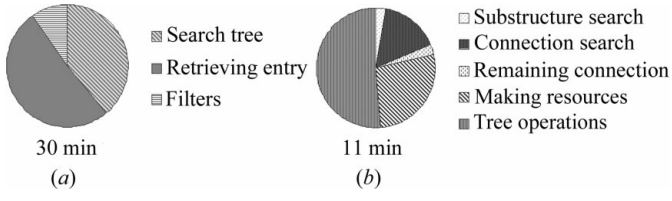

Figure 5
(*a*) A breakdown of the search times for the *T*6(2) motif. (*b*) A further breakdown of search-tree times. A significant proportion of time is spent carrying out tree operations, a fact that reflects the sizes of the search trees produced by the *T*6(2) motif.

3.1. A search for a hydrogen-bonded water pattern

The first search is for a hydrogen-bond motif composed of eight water molecules and six atoms that are allowed to match any atomic species. This search and related searches have been carried out for a separate study to investigate the role of water in crystal structures of organic molecular crystals (Infantes & Motherwell, 2002; Infantes *et al.*, 2003). The water molecules and general atoms are arranged to form a tape motif composed of three connected hexagons. This motif is shown in Fig. 4(*a*) and can be classed as *T*6(2) using the Infantes nomenclature. The regularity of the hexagons is enforced by specifying 'short' and 'long' connections between the substructures. Fig. 4(*b*) shows the CSD entry, refcode AMIMZC10, that contains the motif. The purple lines indicate the query connections. This query is composed of 14 substructures and 44 connections and gives rise to search trees with potential depths of (2 × No. of substructures − 1) = 27.

116 CSD entries were found to contain the motif, the first five entries being ALACUH, AMIMZC10, ARCMPH, ARCMPH01 and AZTHPN. The total time taken to complete the search was 30 min. This timing information is very encouraging and demonstrates the ability of the 3*DSEARCH* algorithm to identify challenging queries that contain many intermolecular connections.

Timing information for this search is shown in Fig. 5(*a*). It can be seen that a significant proportion of the time is spent unpacking and constructing the three-dimensional coordinate information from the CSD. The actual time spent in the search tree is 11 min. A further breakdown of the search-tree times is shown in Fig. 5(*b*). It can be seen that over 50% of the time is spent carrying out tree operations. These include tasks such as deciding which substructure to search for next, checking whether atoms in the structure already form part of the current query match, managing query matches (*i.e.* building up and tearing down matches) and copying matches for storage. Time is also spent in the lookup and retrieval of stored contact and substructure searches. The task 'making resources' in Fig. 5(*b*) refers to the initial construction of a unit cell of coordinates, once and for

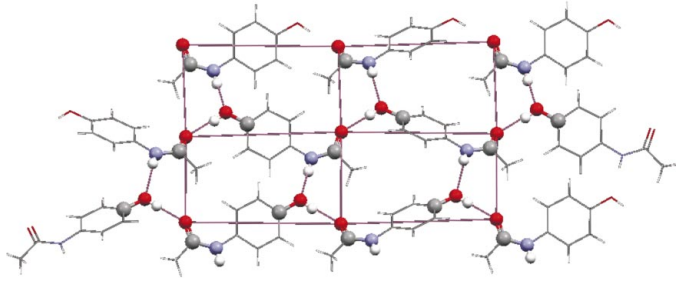

Figure 6
A search query containing 15 substructures (represented in ball-and-stick mode) and 24 connections (purple lines). The structure shown is HXACAN, the orthorhombic form of paracetamol.

all, at the start of the search, which provides a resource for connection searches. The relatively short time spent searching for substructures and connections reflects the fact that the query is composed of repeating substructures and connections, as is typically the case when searching the CSD for a chemical query, and the algorithm can take advantage of this behaviour. In addition, the sizes of the substructures for this example are very small and do not tax the Ullmann algorithm in any way. The time taken to search through the 116 structures that contain the motif is just 9 s.

Such timing information shows that searches that produce large search trees can be carried out on hundreds of thousands of structures in reasonable times.

3.2. A motif containing longer intermolecular connections

As a second example consider the motif shown in Fig. 6. This represents a possible application of the program, where we have a hydrogen-bonded network in a particular crystal, HXACAN, and want to find out how many similar networks exist in the CSD. The atoms, shown in ball-and-stick mode, represent query substructures and purple lines represent query connections. In all there are 15 substructures, 24 connections and 12 angle constraints. The angle constraints specify 90 and 180° angles and ensure a flat motif. Tolerances on these angles are specified as ±5°. Short connections, less than the sum of the van der Waals radii, are specified between C═O, NH and OH groups. Long connections of 6 and 8.5 Å are specified, with a tolerance of ±0.5 Å, to represent the specific geometric arrangement of such groups in orthorhombic paracetamol. Angle constraints of 180 and 90° are imposed, with a tolerance of ±10°, to ensure that the motif is planar.

It took 22 min to search 272 000 CSD structures, and 11 entries were identified as containing the motif, the first five being ALXANM01, ALXANM10, HAHXIX, HXACAN and HXACAN08. To search through the 11 structures that contain the motif takes just 4.8 s.

4. Summary

A new search algorithm has been developed that can identify in an efficient and accurate manner challenging extended chemical queries that are beyond the capabilities of the search engine implemented in *ConQuest*. The algorithm combines graph-matching and contact search techniques into a depth-first backtracking algorithm. Timing information obtained for two motifs (30 and 17 min) demonstrates the effectiveness of this approach in identifying extended queries that contain several substructures and several long contacts. Search times can be expected to reduce significantly by combining the three-dimensional search algorithm with efficient pre-screen steps.

JAC acknowledges Pfizer for financial support.

References

Allen, F. H. (2002). *Acta Cryst.* B**58**, 380–388.
Bruno, I. J., Cole, J. C., Edgington, P. R., Kessler, M., MacRae, C. F., McCabe, P., Pearson, J. & Taylor, R. (2002). *Acta Cryst.* B**58**, 389–397.
Infantes, L., Chisholm, J. & Motherwell, S. (2003). *Cryst. Eng. Commun.* **5**, 480–486.
Infantes, L. & Motherwell, S. (2002). *Cryst. Eng. Commun.* **4**, 454–461.
Rollett, J. S. (1965). *Computing Methods in Crystallography*, p. 25. Oxford: Pergamon Press.
Ullmann, J. R. (1976). *J. Assoc. Comput. Mach.* **23**, 31–42.