J02 CAST/CNMR システムにおける CANOST コードの高速生成法

（東大院情報理工[1]，国立情報学研[2]，理研[3]）
○小市俊悟[1]，岩田覚[1]，宇野毅明[2]，越野広雪[3]，佐藤寛子[2]*

1. 目的

CANOST 法[1]は平面構造の規範的表記法であり，TUT-CNMR, TUT-HNMR システム[2]などにおいて使用されている．このオリジナルの CANOST 法では，必ずしも 1 原子 1 コードの対応になっておらず，例えば水素原子はメチル基などの官能基として 1 炭素原子のコードとしてまとめられた形となっている．

しかし，^{13}C-NMR 化学シフト予測システム CAST/CNMR[3]では，1 原子毎に立体化学コードを付与する立体化学コード化法 CAST[4,5,6]と併記する必要性から，オリジナル CANOST 法を，1 原子 1 コードの平面構造表記法に変更した改良 CANOST 法を検討し，新たなプログラムを作成した．

CANOST 法は分子内の各原子を始点とした分子ツリーを基にコード化を行うが，CAST/CNMR システムで当初開発した本改良 CANOST プログラムでは，始点の原子からの全ての経路を 1 本ずつ探索した後に，規範化規則に従って並べ替え処理を行っていた．つまり，一経路の探索毎に始点原子に戻る形での深さ探索処理を行っていた．このため，CANOST コード化処理に時間を要し，システムを実行する際の律速となっていた．

そこで今回，CAST/CNMR システムの分子コード化処理のうち，平面構造コード化に相当する改良 CANOST コードの高速化を目的として，その全面的な見直しと開発を行った．

2. 改良 CANOST コードの高速生成法

改良 CANOST コードは以下の手順で高速に生成される．

まず，平面分子構造を，原子を頂点，結合を枝とした分子グラフとみなす．この分子グラフに対し，改良 CANOST コードで先頭とする頂点を始点として幅優先探索を行い，層別ネットワークを得る．次に，層別ネットワークに対して規範化のための変形を行う．この規範性が改良 CANOST コードの規範性として反映される．最後に，変形後の層別ネットワークから分子ツリーを作成し，これを線形表記することで，改良 CANOST コードを得る．

上記処理を分子内の全原子について，各原子を層別ネットワークの始点として繰返し実行する．

高速化のポイントは分子グラフを表すのに隣接リストを用いたこと，処理過程において層別ネットワークを用いたこと，その構築のために幅優先探索を採用したこと，規範化手続きの際の順位付けに tuple とよばれる指標を用いたこと，さらに，その処理においてラディックスソートを用いたことである．

以下に各処理について説明する．

2.1 分子グラフのデータ構造

分子グラフを表すデータ構造には隣接リストを利用する．隣接リストとは頂点を表す配列の各要素に，その頂点を端点の一つとする枝のリストを持たせたものである．このデータ構造によって，各頂点に対し，その頂点と一つの枝で接続する，つまり隣接する頂点を容易にかつ高速に探索できる．

2.2 層別ネットワークの作成

上記分子グラフ内の頂点の 1 つを始点とすると，始点とその他の各頂点とを結ぶ枝数最小の道が定義できる．等しい最小枝数を有する頂点を 1 つの層として分類し，最小枝数順に並べたものが層別ネットワークである．

図 1-a,b にそれぞれ分子グラフと層別ネットワークの例を示す．

分子グラフからこの層別ネットワークを構築するには，幅優先探索法を利用する．幅優先探索法はグラフの線形時間探索法の一つである．したがって，分子グラフの頂点数を n，枝数を m として，$O(n+m)$の計算時間で分子グラフから層別ネットワークを構築できる．

2.3 層別ネットワークの規範化

層別ネットワークの規範化とは，同一の分子グラフであれば，異なる描かれ方をされていても同一のネットワークとなるように，ある規則に従って各層内の並べ替えを行うことである．

この規範化は頂点に順位を付けることで実現される．各層内の頂点間の順位はその頂点の持つ tuple によって決められる．アルゴリズムは tuple の更新とそれによる順位付け

*cheminfo@nii.ac.jp

という繰返しを含み，その繰返しにより徐々に順位を確定していく．

Tuple とは，(1)親（1 つ上の層で，着目している頂点と隣接している頂点のこと）の中で

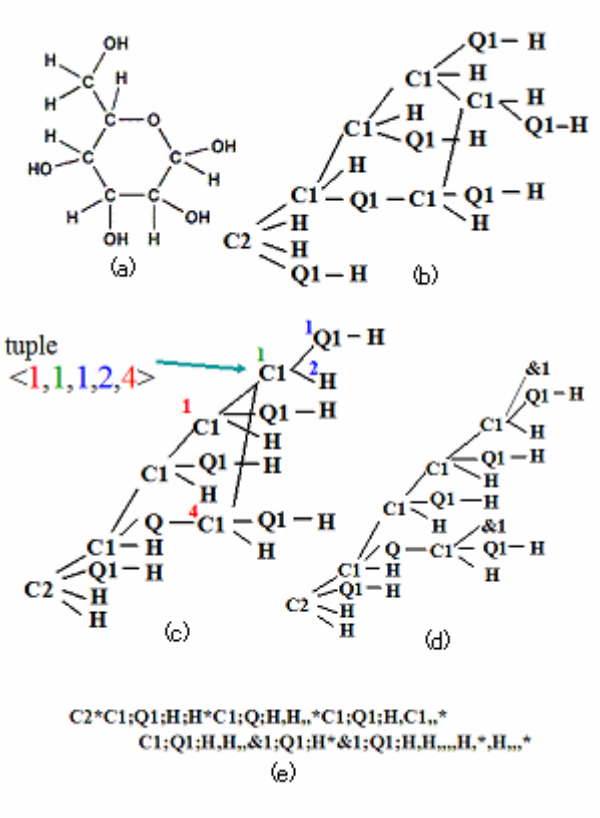

図 1　改良 CANOST コードの高速生成

最高位のものの順位, (2)着目している頂点の順位，(3)子（1 つ下の層で，着目している頂点と隣接している頂点のこと）の中で *i* 番目のものの順位(*i*=1, 2,..., *k* (*k*: 子の頂点の数))，(4)親の中で *j* 番目のものの順位(*j*=2, 3,..., *l* (*l*: 親の頂点の数))を順に並べたリストである．

Tuple の辞書式比較はラディックスソートによって行う．そして，その更新と順位付けを，最低位から最高位へのボトムアップと最高位から最低位へのトップダウンの方向に交互に繰返すことにより順位を確定する．Tuple 中の順位は，繰り返しの 1 回目においては，各頂点の CANOST コードの順位であり，それ以降は繰り返し中の，その時点での順位である．

図 1-c は図 1-b を規範化したものである．また, 矢印で示した頂点の具体的な tuple を 1 つ示してある．

2.4　分子ツリーと改良 CANOST コード

分子ツリーは層別ネットワーク中の環構造を切断し，その箇所に&*s*（*s* は切断箇所にふられるシリアル番号）コードをふることにより生成される（図 1-d）．

得られた分子ツリーを同義の線形表記に変換しつつ出力することで，改良 CANOST コードを効率的に生成・出力する（図 1-e）．

3．結果

計算時間の測定結果を表 1 に示す．分子内の全原子を始点とする全コードを生成するのに要したユーザー計算時間である．

繰返し構造を持たない天然有機化合物の中では最大の maitotoxin（492 原子）でも 2.3 秒で処理が終了しており，実用性に適う速度が実現できたことがわかる．

表 1　計算時間の測定結果

分子名	原子数	計算時間（秒）
hemibrevetoxin B	77	0.07
20-hydroxyecdysone	78	0.06
aflastatin A	202	0.35
palytoxin	409	1.50
maitotoxin	492	2.30

Pentium M 1.3GHz による

4．まとめ

グラフ理論のアルゴリズムを応用することで，改良 CANOST コードの高速な生成法の開発に成功し，CAST/CNMR の実用化を促進することができた．

Tuple の比較と更新の繰返し処理が，本高速改良 CANOST コード化の律速であるが，個々の処理が高速に実行されるために，コード化の全処理が高速のうちに実行される．また，本手法は多項式時間アルゴリズムではないので，可能なグラフ構造が制限のある分子構造に限定されたことは，実用的な高速化が実現できた一因といえる．

現在，より利用目的に則した線形表記に対応するための改良を進めている．

参考文献

[1] H. Abe, Y. Kudo, T. Yamasaki, K. Tanaka, M. Sasaki, S. Sasaki, *J. Chem. Inf. Comput. Sci.*, **24**, 212-216(1984)

[2] K. Funatsu, S. Sasaki, *J. Chem. Inf. Comput. Sci.* **36**, 190-204(1996)

[3] H. Satoh, H. Koshino, J. Uzawa, T. Nakata, *Tetrahedron*, **59**/**25**, 4539-4547(2003)

[4] H. Satoh, H. Koshino, K. Funatsu, T. Nakata, *J. Chem. Inf. Comput. Sci.*, **40**, 622-630(2000)

[5] H. Satoh, H. Koshino, K. Funatsu, T. Nakata, *J. Chem. Inf. Comput. Sci.*, **41**, 1106-1112(2001)

[6] H. Satoh, H. Koshino, T. Nakata, *J. Comput Aided Chem.*, **3**, 48-55(2002)