幅優先探索順による木状化学構造の並列列挙

Parallelization of enumerating tree-like chemical compounds by breadth-first search order

化学研究所 バイオインフォマティクスセンター 数理生物情報研究領域　林田守広

背景と目的

新薬候補の探索や，質量分析器からの化学構造決定，原子数などの制約条件下での化学構造空間の分析などのために，化学構造の列挙が重要な技術の一つになっている．本研究では，目的の化合物に含まれる各原子の数を入力として，環を持たない木状の化学構造を重複なくすべて列挙する問題を扱う．先行研究では単一プロセスで動作する，幅優先探索順によるノード追加のアルゴリズムを開発した．本研究では，このアルゴリズムを拡張しさらに高速化するために，複数のプロセスで並列に動作する三つのアルゴリズム BfsEnumP1-3 を提案した[1].

検討内容

まず初めに元となる，先行研究での列挙アルゴリズム BfsSimEnum，BfsMulEnum を説明する．入力は原子の種類 $\Sigma=\{l_i\}$と各原子の価数 $val(l_i)$と個数 n_{li} とする．この条件を満たす木状化学構造を重複なくすべて列挙する．例えば，$\Sigma=\{C, O, H\}$, $val(C)=4$, $val(O)=2$, $val(H)=1$, $n_C=6$, $n_O=2$, $n_H=14$，つまり $C_6O_2H_{14}$ などを入力とする．
化学構造は根付き順序木として表現され，各ノードには Σ中の原子のラベルが付与される．また共有結合の多重度は辺の多重度によって表現される．化学構造の生成は，ノードを一つも持たない木から始めて，幅優先探索順にノードを追加していく．BfsSimEnum は単純辺のみを持つ木構造を列挙し，多重辺は考慮しない．BfsMulEnum は多重結合を持つ化合物の場合に，BfsSimEnum の出力をもとに単純辺を多重辺に置き換えることで列挙を行う．価数が1の原子は最後に付加することで最終的な出力とする．

列挙において重複する構造の生成を避けるために木グラフに対する標準形を定義する．原子のラベルに順序(例えば，C > O > H)を設けることで，ラベル付けされた根付き順序木 T_1, T_2 に $T_1 > T_2$ のような順序を定める．ある根付き順序木 T について，任意の兄弟ノード v_1, v_2 の間で，そのノードを根とする部分木 $T(v_1)$, $T(v_2)$ の間に $T(v_1) > T(v_2)$の関係が成り立てば，left-heavy と呼ぶ．また木の直径となるパスの中央のノードまたは辺のどちらかの端点に根が位置するときに center-rooted と呼ぶ．新たなノードの追加においては，追加後の木が必ず left-heavy かつ center-rooted となる木のみを生成する．標準形はさらに r を根とするとき，直径となるパスの中央が辺(r,v)であるときにはこの辺を除いた二つの部分木が $T(v)\geq T(r)$ を満たすときに標準形であると定義する．

探索過程を表現する，化学構造をノードする木を family 木とよぶ．BfsSimEnum における原子の追加によって，ある化学構造から別の化学構造が生成されるとき，family 木ではそれらのノードの間に辺が追加される．family 木の根は原子 0 個の木とする．本研究では，この family 木を分割し各プロセスに割り当てることを考える．family 木の深さが浅いときには分割できるだけのノードが無いので，ある深さ d まではすべてのプロセスが独立に family 木を形成するとする．深さ d 以降は，深さ d の各ノードを各プロセスへ重複なくすべて割り当てそれぞれのプロセスで処理する．割り当て方によって以下の三つのアルゴリズムを提案する．ここで使用するプロセスの数を N とし，プロセス 0 からプロセス N-1 までに割り当てるとする．

I. BfsEnumP1

深さ d のノードに BfsSimEnum で生成される順番に番号 c を付ける. c を N で割った余りを p とすると, 番号 c のノードをプロセス p に割り当てる. プロセス p はノード c を根とする family 木の部分木を形成し, プロセス p 以外はこの部分木の形成を省略し, 深さ d 以下の次のノードを生成する. 各プロセスは深さ d のすべてのノードを生成し, 自プロセスの割り当て分が列挙完了すると終了する.

II. BfsEnumP2

深さ d の各ノードから形成される family 木の部分木の大きさには偏りがある. BfsEnumP1 では, 小さい部分木だけが一つのプロセスに割り当てられ, 計算資源を有効に活用できない場合が考えられる. 各プロセスでの処理時間はできるだけ均等になるのが望ましい. そこで BfsEnumP2 では, 部分木の大きさ cost をノード c の化学構造から推定し, 各プロセスで cost の和ができるだけ均等になるように割り当てる. deg(v)を v の次数とし, cost は $val(l(v_i))-deg(v_i)$の追加可能な v_i での和と残りの原子数の重み付き和で定義する. 深さ d のノードに到達するごとに cost を計算し, プロセスごとのこれまで処理したノードの cost の和が最小であるプロセスに割り当てる.

III. BfsEnumP3

上の二つの割り当てはプロセス間の通信がなく静的に割り当てが決まっていたが, BfsEnumP3 では一つ余分なプロセスが割り当てを管理し, 他のプロセスからの割り当て要求があるごとに動的に割り当てを決定し, 通知する. 深さ d の各ノードに対して, 最初に到達したプロセスがそのノードを担当する.

結果と考察

$C_{26}H_{54}$, $C_{16}O_4H_{34}$, $C_{10}N_3O_2H_{25}$, $C_{20}H_{40}$, $C_{12}O_4H_{16}$, $C_{11}N_3O_2H_{21}$ のそれぞれの入力について, パラメーターd を 4 から 8 までとプロセス数を 1 から 12 まで変更し, 計算時間の計測を行った. 列挙数はそれぞれ, 93839412, 278960984, 29105924, 4224993, 282338151, 7268812476 であった. 殆どの場合に BfsEnumP2 が BfsEnumP1 よりも計算時間が短く, さらに BfsEnumP3 がこれらよりも勝っており, 12 プロセス使用時に単一プロセスの場合の約 10 パーセントの実行時間を達成した.

パラメーターd については, $C_{26}H_{54}$ を入力としたときの family 木の深さ 4 におけるノード数は 4 であり, d=4 を入力とするとプロセス数が 5 以上のときには, 割り当てられないプロセスが生じる. 従ってある程度大きな d が必要である. 逆に, $C_{10}N_3O_2H_{25}$ を入力としたときの family 木の深さ 8 でのノード数は 50522 であり, 計算時間は BfsEnumP3 が BfsEnumP1-2 よりも長かった. BfsEnumP1-2 では数が多い程各プロセスでの処理時間が平均化する一方, BfsEnumP3 では管理のための通信に時間がかかり遅くなったと考えられる. 本研究での実験結果からは使用するプロセス数の約 100 倍のノード数となるように d を設定すると良好な計算時間が得られることが示唆された.

今後の課題としては, BfsEnumP2 の cost の計算方法の改良による高速化, また列挙可能な化学構造の拡大, 例えばベンゼン環やナフタレン環を含む構造の列挙手法の並列化がある.

参考論文

1. M. Hayashida, J. Jindalertudomdee, Y. Zhao, T. Akutsu, Parallelization of enumerating tree-like chemical compounds by breadth-first search order, *The 8th International Conference on Systems Biology*, 2014.