Parallelization of Enumerating Tree-like Chemical Compounds by Breadth-first Search Order

Morihiro Hayashida[1,a)] Jira Jindalertudomdee[1,b)] Yang Zhao[1,c)] Tatsuya Akutsu[1,d)]

Abstract: Enumerating chemical compounds greatly assists designing drugs and determining chemical structures from mass spectrometry. In our previous study, we developed efficient algorithms, *BfsSimEnum* and *BfsMulEnum* for enumerating tree-like chemical compounds without and with multiple bonds, respectively.
In this technical report, we develop a parallelized algorithm *BfsEnumP* by modifying BfsSimEnum in a simple manner to further reduce execution time. BfsSimEnum constructs a family tree in which each node denotes a molecular tree. BfsEnumP divides the family tree at some depth, and each subtree to be constructed is assigned to a processor. For evaluation, we perform experiments for several instances with varying the division depth and the number of processors, and show that it achieves about 80% parallelization efficiency.

1. Introduction

Enumerating chemical compounds assists designing drugs and determining chemical structures from mass spectrometry. Hence, algorithms and mathematical models for the enumeration have been developed. A chemical compound is often represented as a molecular graph for the enumeration, which is defined as a connected graph with vertices labeled by atomic symbols and multi-edges labeled by chemical bonds. Here, the degree of a vertex means the valence of the atom and the multiplicity of a multi-edge means the bond order. Given chemical formula and some restrictions, chemical structures desired for biological system are enumerated by constructing all distinct graph structures. MOLGEN has been developed over two decades [1], [2], and becomes a popular enumeration tool. EnuMol enumerates tree-like chemical compounds, or molecular tree graphs, by depth-first search (DFS) order [3], [4], [5]. In our previous study, we developed efficient algorithms *BfsSimEnum* and *BfsMulEnum* for enumeration of tree-like chemical compounds by breadth-first search (BFS) order [6]. For many instances, execution times by BfsSimEnum and BfsMulEnum were shorter than or comparable to those by the DFS-type method.

In this technical report, we propose a parallelized algorithm *BfsEnumP* by modifying BfsSimEnum. BfsSimEnum constructs a family tree in which each node denotes a molecular tree. BfsEnumP divides the family tree at some depth, and each subtree to be constructed is assigned to a processor. We perform computational experiments for several instances with varying the division depth and the number of processors, and show that it achieves about 80% parallelization efficiency.

[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611–0011, Japan
[a)] morihiro@kuicr.kyoto-u.ac.jp
[b)] jira@kuicr.kyoto-u.ac.jp
[c)] tyoyo@kuicr.kyoto-u.ac.jp
[d)] takutsu@kuicr.kyoto-u.ac.jp

2. Preliminaries

2.1 Enumeration problem

A molecular tree can be represented as a rooted ordered tree $T(V, E)$ with a set V of vertices and a set E of single and multiple edges, where each vertex corresponds to an atom, and each edge corresponds to a covalent bond. Let $\Sigma = \{l_1, l_2, \ldots, l_s\}$ be a set of labels representing distinct atoms. Let $val(l_i)$ be the valence of the atom corresponding to l_i. Let $num_T(l_i)$ be the number of vertices labeled as l_i in T. Let $l(v)$ and $degree(v)$ be the label, and degree of vertex v in T, respectively. Then, we define the tree-like compound enumeration problem as follows.

Problem 1 Given a set Σ of labels, the valence $val(l_i)$ and number n_{l_i} of each label l_i, enumerate all molecular trees T without any redundancy such that $num_T(l_i) = n_{l_i}$ for all $l_i \in \Sigma$ and $degree(v) = val(l(v))$ for all $v \in T$.

2.2 Family tree

Our approach searches a special tree structure called *family tree*. Figure 1 shows the family tree for $C_2O_2H_2$ by BfsSimEnum and BfsMulEnum, where hydrogen atoms are added at the end of enumeration. Each vertex of the family tree represents a molecular tree. We call a molecular tree *center-rooted* if its root is the center vertex or an endpoint of the center edge of a path with the maximum length. We introduce a total order to Σ, for example, C > O > H for $\Sigma = \{C, O, H\}$. We call a molecular tree *left-heavy* if a left sibling of each vertex is larger than and equal to the vertex (see [6] for the rigorous definition). Then, BfsSimEnum always generates left-heavy and center-rooted trees with labeled vertices to reduce the search space. Finally, a generated tree is discarded if it is not in normal form [6]. It should be noted that molecular

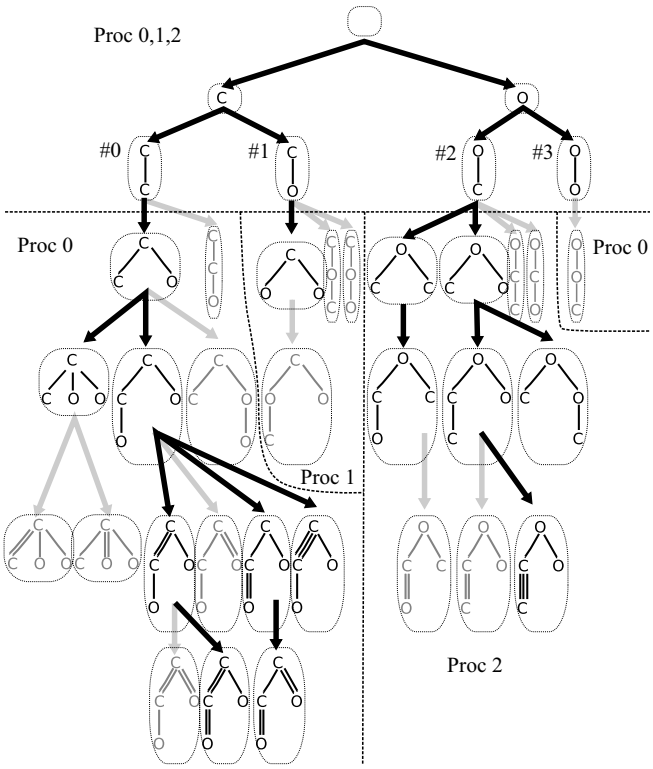

Fig. 1 Illustration of BfsEnumP with division depth 2 for $C_2O_2H_2$ using 3 processors. Molecular trees in gray color are regarded as invalid by the algorithms. It should be noted that hydrogen atoms are added as leaves at last.

trees are generated by BFS order while a family tree is searched by DFS order.

3. Methods

We propose a parallelized algorithm BfsEnumP by modifying BfsSimEnum in a simple manner. In growing a family tree, BfsSimEnum adds an atom to a molecular tree by BFS order. In BfsEnumP, given the numbers of atoms n_{l_i} for $l_i \in \Sigma$ and some parameters, division depth d, processor identifier p, and the number of processors N, each processor runs as follows.

Step 1. constructs a family tree up to depth d, and assign numbers i to the vertices (molecular trees) in the depth by BFS order.

Step 2. for each vertex i, constructs the subtree of the family tree from the vertex if the remainder is equal to p when i is divided by N. Otherwise, skip it.

Figure 1 illustrates the procedure with division depth 2 for $C_2O_2H_2$ using 3 processors. In depth 2 of the family tree, there are 4 vertices (molecular trees with 2 atoms) in the family tree, and the construction from each vertex is assigned to each processor, that is, vertices $0, 1, 2, 3$ are assigned to processors $0, 1, 2, 0$, respectively. It should be noted that message passing between processors does not occur in the construction of a family tree.

4. Results

We employed PrimeHPC FX10 with Fujitsu SPARC64 IXfx processors for evaluation of our proposed method. BfsEnumP was implemented in C++ using MPI library.

Table 1 shows the results on elapsed times (seconds) of BfsEnumP with division depth $d = 7$ and 8 for $C_{26}H_{54}$ and $C_{12}O_4H_{16}$ using up to 8 processors. For $C_{12}O_4H_{16}$, the elapsed time for $d = 8$ was shorter than that for $d = 7$. It is considered that vertices were assigned in a better balanced manner for $d = 8$ than for $d = 7$ because the number of vertices in depth 8 was larger than that in depth 7. On the other hand, that was not necessarily so for $C_{26}H_{54}$, and the elapsed times for $d = 8$ using 2,3,6,7 processors were longer than those for $d = 7$.

Table 1 Elapsed time (seconds) of BfsEnumP with division depth $d = 7$ and 8 for $C_{26}H_{54}$ and $C_{12}O_4H_{16}$ using 1,...,8 processors.

#processors	$C_{26}H_{54}$		$C_{12}O_4H_{16}$	
	$d = 7$	8	7	8
1	328.50		727.50	
2	179.13	203.41	359.21	349.87
3	157.67	160.50	274.14	235.01
4	139.51	136.04	190.71	183.33
5	86.94	80.70	153.08	145.92
6	89.72	107.87	141.62	135.91
7	60.94	71.42	129.07	109.16
8	84.26	79.10	124.06	111.42

The parallelization efficiency is defined as $T_1/(N \cdot T_N)$, where T_N denotes the elapsed time by N processors. Then, we can see from the table that BfsEnumP achieved about 80% parallelization efficiency when $d = 8$ by 8 processors for $C_{12}O_4H_{16}$.

5. Conclusion

In this technical report, we proposed BfsEnumP for enumerating tree-like compounds in parallel by modifying our previous method BfsSimEnum. We performed experiments for several instances with varying parameters and the number of processors, and BfsEnumP achieved about 80% parallelization efficiency. It, however, is necessary to improve the proposed method because the assignment to processors was not sufficiently balanced.

Moreover, it is important to deal with more complex ring structures. Extensions toward enumerating general compounds and combination with biological properties for BfsSimEnum and BfsEnumP should be another future work.

Acknowledgments

This work was partially supported by Grants-in-Aid #22240009, #24500361, and #25-2920 from MEXT, Japan, and also by MEXT SPIRE Supercomputational Life Science.

References

[1] Gugisch, R., Kerber, A., Kohnert, A., Laue, R., Meringer, M., Rucker, C. and Wassermann, A.: *Molgen 5.0, a Molecular Structure Generator*, Bentham Science Publishers Ltd. (2012).

[2] Faulon, J. L., D. P. Visco, J. and Rose, D.: Enumerating molecules, *Reviews in Computational Chemistry*, Vol. 21, pp. 209–286 (2005).

[3] Ishida, Y., Kato, Y., Zhao, L., Nagamochi, H. and Akutsu, T.: Branch-and-bound algorithms for enumerating treelike chemical graphs with given path frequency using detachment-cut, *Journal of Chemical Information and Modeling*, Vol. 50, No. 5, pp. 934–946 (2010).

[4] Fujiwara, H., Wang, J., Zhao, L., Nagamochi, H. and Akutsu, T.: Enumerating treelike chemical graphs with given path frequency, *Journal of Chemical Information and Modeling*, Vol. 48, No. 7, pp. 1345–1357 (2008).

[5] Shimizu, M., Nagamochi, H. and Akutsu, T.: Enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies, *BMC Bioinformatics*, Vol. 12, No. Suppl 14, pp. 1–9 (2011).

[6] Zhao, Y., Hayashida, M., Jindalertudomdee, J., Nagamochi, H. and Akutsu, T.: Breadth-first search approach to enumeration of tree-like chemical compounds, *Journal of Bioinformatics and Computational Biology*, Vol. 11, p. 1343007 (2013).