

2-Branch構造に基づく非環式化学グラフの 生成プログラムの使用説明書

2020 年 8 月 19 日

目 次

1	概要	1
2	用語の説明	4
3	プログラムの入力と出力	4
3.1	プログラムの入力	4
3.2	入力データの形式	5
3.3	プログラムの出力	6
3.4	出力データの形式	7
4	プログラムの実行と計算例	7
4.1	実行方法	7
4.2	計算例	7
	参考文献	9

1 概要

この冊子では、2-branch 構造 [1] に基づく非環式化学グラフの生成アルゴリズムを実装した二つのプログラムについて説明する。一つのプログラムでは、入力の特徴ベクトルに対して、その特徴ベクトルを持つ 2-branch-number が 2 の非環式化学グラフを出力する。もう一つのプログラムでは、入力の特徴ベクトルを持つ 2-branch-number が 4 の非環式化学グラフを出力する。

また、このフォルダにはこの冊子、README.txt と License.txt 以外に以下のファイルとフォルダが含まれている。

- フォルダ 2.2-branches

2-branch-number が 2 の非環式化合物を生成するためのプログラムのフォルダである。

- フォルダ include

このプログラムに使われるヘッダファイルのフォルダである。

- * cross.timer.h

- 計算時間を計測するためのヘッダファイルである。

- * data_structures.hpp

- 化学グラフのデータ構造を定義するヘッダファイルである。

- * debug.h

- デバッグのためのヘッダファイルである。

- * fringe_tree.hpp

- ヘリ木 (fringe tree) [1] を列挙する関数のヘッダファイルである。

- * tools.hpp

- 便利なツール関数のヘッダファイルである。

- フォルダ instances

入力のインスタンスのフォルダである。

- * ha_tv3500_n15_dia10_dmax3_k2_bn2_bh1.txt

- 原子化熱 (Heat of Atomization) の活性値が 3500, 節点数が 15, 直径が 10, 次数上限が 3, branch-parameter が 2, 2-branch-number が 2, 2-branch-height が 1 となる特徴ベクトルが記載されている入力データである。

- * ha_tv4500_n20_dia12_dmax4_k2_bn2_bh1.txt

- 原子化熱の活性値が 4500, 節点数が 20, 直径が 12, 次数上限が 4, branch-parameter が 2, 2-branch-number が 2, 2-branch-height が 1 となる特徴ベクトルが記載されている入力データである。

- * ha_tv7000_n30_dia15_dmax3_k2_bn2_bh1.txt

- 原子化熱の活性値が 7000, 節点数が 30, 直径が 15, 次数上限が 3, branch-parameter が 2, 2-branch-number が 2, 2-branch-height が 1 となる特徴ベクトルが記載されている入力データである。

- * ha_tv8500_n40_dia30_dmax4_k2_bn2_bh1.txt

- 原子化熱の活性値が 8500, 節点数が 40, 直径が 30, 次数上限が 4, branch-parameter が 2, 2-branch-number が 2, 2-branch-height が 1 となる特徴ベクトルが記載されている入力データである。

* `ha_tv11000_n50_dia40_dmax3_k2_bn2_bh1.txt`

原子化熱の活性値が11000, 節点数が50, 直径が40, 次数上限が3, branch-parameter が2, 2-branch-number が2, 2-branch-height が1 となる特徴ベクトルが記載されている入力データである.

– `main.cpp`

2-branch-number が2 の非環式化合物を生成するためのプログラムである.

● フォルダ `4_2-branches`

2-branch-number が4 の非環式化合物を生成するためのプログラムのフォルダである.

– フォルダ `include`

このプログラムに使われるヘッダファイルのフォルダである.

* `cross_timer.h`

計算時間を計測するためのヘッダファイルである.

* `data_structures.hpp`

化学グラフのデータ構造を定義するヘッダファイルである.

* `debug.h`

デバッグのためのヘッダファイルである.

* `fringe_tree.hpp`

ヘリ木 (fringe tree) を列挙する関数のヘッダファイルである.

* `tools.hpp`

便利なツール関数のヘッダファイルである.

– フォルダ `instances`

入力のインスタンスのフォルダである.

* `ha_tv3500_n15_dia10_dmax3_k2_bn4_bh2.txt`

原子化熱の活性値が3500, 節点数が15, 直径が10, 次数上限が3, branch-parameter が2, 2-branch-number が4, 2-branch-height が2 となる特徴ベクトルが記載されている入力データである.

* `ha_tv4500_n20_dia12_dmax4_k2_bn4_bh2.txt`

原子化熱の活性値が4500, 節点数が15, 直径が10, 次数上限が4, branch-parameter が2, 2-branch-number が4, 2-branch-height が2 となる特徴ベクトルが記載されている入力データである.

* `ha_tv8000_n30_dia20_dmax3_k2_bn4_bh2.txt`

原子化熱の活性値が8000, 節点数が30, 直径が20, 次数上限が3, branch-parameter が2, 2-branch-number が4, 2-branch-height が2 となる特徴ベクトルが記載されている入力データである.

* `ha_tv8000_n35_dia20_dmax4_k2_bn4_bh2.txt`

原子化熱の活性値が8000, 節点数が40, 直径が30, 次数上限が4, branch-parameter が2, 2-branch-number が4, 2-branch-height が2 となる特徴ベクトルが記載されている入力データである.

* `ha_tv8500_n40_dia20_dmax3_k2_bn4_bh2.txt`

原子化熱の活性値が8500, 節点数が50, 直径が40, 次数上限が3, branch-parameter が2, 2-branch-number が4, 2-branch-height が2 となる特徴ベクトルが記載されている入力データである.

— `main.cpp`

2-branch-number が4 の非環式化合物を生成するためのプログラムである.

次に, この冊子の構成について説明する. 第2節では, この冊子およびプログラム内で使用している用語について説明する. 第3節では, プログラムの入力と出力について説明する. 第4節では, 実際の計算例を示す.

2 用語の説明

この節では、用語について説明する。

- 特徴ベクトル

各種類の元素の原子数等の化学物質を説明する数値やグラフの直径等の化学グラフのトポロジーに基づいて計算される数値のベクトル。これら二つのプログラムに使われる特徴ベクトルの内容は3.2に具体例を用いて説明する。

- 非環式化学グラフ

化学グラフは節点集合と枝集合の組で化合物の構造を表すものである。各節点には原子の種類が割り当てられており、各枝には結合の多重度が割り当てられている。非環式化学グラフは環が一つもない分子構造の化合物を表すものである。この冊子では水素原子が省略された化学グラフを扱っていく。

- branch-parameter, k -branch, k -branch-number, k -branch-height, ヘリ木 (fringe tree), 内部・外部節点 (internal/external vertex), 内部・外部原子結合 (internal/external adjacency-configuration), 内部・外部次数結合 (internal/external bond-configuration) について参考文献 [1] に参考すること。

3 プログラムの入力と出力

この節では、プログラムの入力と出力について説明する。3.1 節では、プログラムの入力情報について説明する。3.2 節では、計算機上で入力する際のデータ形式について説明する。3.3 節では、プログラムの出力情報について説明する。3.4 節では、計算機上でプログラムを実行した際に出力されるデータ形式について説明する。

3.1 プログラムの入力

これら二つのプログラムでは四つの情報を入力とする。一つ目は特徴ベクトルが記入されたテキストファイルである。このテキストファイルのフォーマットは3.2節に示す。二つ目はプログラムの計算時間の上限秒である。三つ目は出力する化学グラフの個数の上限である。四つ目は出力される化学グラフを保存する SDF ファイル名である。

3.2 入力データの形式

この節では、3.1 節に説明していた入力ファイルの形式について説明する．入力するファイルの具体例を以下に示す．

入力データの形式

```
2
C 120 4 5 8
0 160 2 2 0
3
C C 2 0 3
C 0 1 4 0
C C 1 2 5
0 6
4 1
3 1
0 0
10
3
1 2 1 0 1
1 2 2 0 0
1 2 3 0 0
1 3 1 0 2
1 3 2 0 3
1 4 1 0 0
2 2 1 2 0
2 2 2 0 0
2 2 3 0 0
2 3 1 4 1
2 3 2 0 0
2 4 1 0 0
3 3 1 0 1
3 3 2 0 0
3 4 1 0 0
4 4 1 0 0
```

この具体例を用いて、各行の内容を説明する．数値例とそれぞれの内容の対応を表 1 に示す．

表 1: 入力ファイルの構成

数値例	内容
2	原子の種類
C 120 4 5 8 O 160 2 2 0	それぞれの原子シンボル, 質量の十倍, 価数, 原子の内部節点数, 原子の外部節点数
3	原子結合の種類
C C 2 0 3 C O 1 4 0 C C 1 2 5	それぞれの原子結合の原子, 原子, 多重度, 内部原子結合の数, 外部原子結合の数
0 6 4 1 3 1 0 0	次数が1の内部節点数, 次数が1の外部節点数 次数が2の内部節点数, 次数が2の外部節点数 次数が3の内部節点数, 次数が3の外部節点数 次数が4の内部節点数, 次数が4の外部節点数
10	直径
3	次数の上限
1 2 1 0 1 1 2 2 0 0 1 2 3 0 0 1 3 1 0 2 1 3 2 0 3 1 4 1 0 0 2 2 1 2 0 2 2 2 0 0 2 2 3 0 0 2 3 1 4 1 2 3 2 0 0 2 4 1 0 0 3 3 1 0 1 3 3 2 0 0 3 4 1 0 0 4 4 1 0 0	それぞれの次数結合の次数, 次数, 多重度, 内部次数結合の数, 外部次数結合の数

3.3 プログラムの出力

これら二つのプログラムの出力は, 入力された特徴ベクトルを持つ化学グラフの個数の下限, 生成された化学グラフの個数, プログラムの計算時間と入力された特徴ベクトルを持つ化学グラフである. 化学グラフはファイル名が入力された SDF ファイルに出力される.

3.4 出力データの形式

これら二つのプログラムの出力する SDF ファイルのフォーマットについて、<https://www.chem-station.com/blog/2012/04/sdf.html> などの解説が分かりやすい。さらに、正確な定義書として、公式資料 http://help.accelrys.com/ulm/online/1.0/content/ulm_pdfs/direct/reference/ctfileformats2016.pdf を参照するとよい。

4 プログラムの実行と計算例

ここでは二つのプログラムの実行方法と、具体例を入力した場合の計算結果を示す。

4.1 実行方法

二つのプログラムの実行方法は同じである。

- 環境確認

ISO C++ 2011 標準に対応する C++コンパイラがあれば問題ないと考えられる。Linux Mint 18 & 19, コンパイラ g++ ver 5 & 7 で確認したが, g++がインストールされていない場合は, 次のようにインストールできる。

```
$ sudo apt install g++
```

- コンパイル

```
$ g++ -o main main.cpp -O3 -std=c++11
```

(g++ 7 の場合は `-std=c++11` を省略できる。)

- 実行

```
$ ./main instance.txt a b output.sdf
```

instance.txt で入力のテキストファイル, a で計算時間の上限, b で出力する化学グラフの個数の上限, output.txt で化学グラフの出力の SDF ファイルを指定する。

4.2 計算例

具体例として, 2-branch-number が 2 のプログラムを以下のような条件で実行する。

- 入力ファイル: フォルダ instances 内の ha_tv3500_n15_dia10_dmax3_k2_bn2_bh1.txt
- 計算時間上限: 10 秒
- 出力する化学グラフの個数上限: 2
- 化学グラフの指定出力ファイル: output.sdf

実行のコマンドは以下になる。

```
./main ../instances/ha_tv3500_n15_dia10_dmax3_k2_bn2_bh1.txt 10 2 output.sdf
```

このコマンドを実行した場合, 以下のような出力結果が得られる。

ターミナルに実行結果の具体例

A lower bound on the number of graphs = 56

Number of generated graphs = 2

Time : 0.00842078s.

output.sdf の内容

1

2-branches

2-branches

15 14 0 0 0 0 0 0 0 0999 V2000

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 2 2 0 0 0 0

1 3 1 0 0 0 0

1 6 1 0 0 0 0

3 4 1 0 0 0 0

3 5 1 0 0 0 0

6 7 1 0 0 0 0

7 8 2 0 0 0 0

7 9 1 0 0 0 0

9 10 1 0 0 0 0

10 11 1 0 0 0 0

11 12 1 0 0 0 0

12 13 2 0 0 0 0

```

12 14 1 0 0 0 0
14 15 1 0 0 0 0
M END
$$$$
2
2-branches
2-branches
15 14 0 0 0 0 0 0 0 0999 V2000
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 2 1 0 0 0 0
1 4 1 0 0 0 0
1 5 1 0 0 0 0
2 3 1 0 0 0 0
5 6 1 0 0 0 0
6 7 2 0 0 0 0
6 8 1 0 0 0 0
8 9 1 0 0 0 0
9 10 1 0 0 0 0
10 11 1 0 0 0 0
11 12 2 0 0 0 0
11 13 1 0 0 0 0
13 14 2 0 0 0 0
13 15 1 0 0 0 0
M END
$$$$

```

参考文献

- [1] N. A. Azam, J. Zhu, Y. Sun, Y. Shi, A. Shurbevski, L. Zhao, H. Nagamochi and T. Akutsu.
A Novel Method for Inference of Acyclic Chemical Compounds with Bounded Branch-height
Based on Artificial Neural Networks and Integer Programming.