6K-06

Median Iterationアルゴリズムのハイパーグラフへの拡張

鈴木 琢人† 山本 幹雄‡
筑波大学 情報メディア創成学類† 筑波大学 システム情報系‡

1 はじめに

TRIE(Fredkin,1960)によるngram言語モデルの効率的な実装法として，ダブル配列(Aoe,1989)を用いたDALM(Double-Array Language Model)(Yasuhara et al.,2016)がある．ダブル配列を構築する際に，各ノードの子ノードのID幅が小さいとコンパクトな配置になる．そこで，ID幅が小さくなるようにIDを入れ替える操作を行うが，この操作はハイパーグラフの線形配置問題として捉えることができる．

本稿では，これを解くために，巨大なデータ構造でも実行可能な，一般グラフの線形配置問題のヒューリスティックアルゴリズムであるMedian Iteration(以下MIとする)(Koren and Harel,2002)を，ハイパーグラフに適用できるように拡張した．また，MIは緩和問題に変換して解くため，元の問題が必ず良くなるとは限らない．我々は，MIで求まる各IDの移動を1対1での入れ替えにすることで，他のIDに影響を与えずに改善する手法を提案する．

本手法を用いた実験では，提案手法がハイパーグラフに拡張したMIより，解を改善することを示した．また，得られたIDを用いてDALMを構築した結果，提案手法で充填率(配列長の効率)が向上することも示した．

2 線形配置問題

グラフの線形配置問題とは，無向グラフの頂点を数直線の整数点に，重複ないように配置した時の，辺の長さの総和を最小化する問題[1]である(Garey and Johnson,1979)．

ハイパーグラフの線形配置問題(Jin et al.,2008)では，ハイパーエッジの接続する頂点が3つ以上の場合もあるため，辺の長さはハイパーエッジの幅に相当する．$V = 1,2,\ldots,n$をn個の頂点集合，Xをハイパーエッジ集合とし，$HG(V,X)$をハイパーグラフとする．ここで線形配置πとは，Vの順列である．頂点iの位置を$\pi(i)$とすると，配置コストは以下のようになり，これを最小化するのがハイパーグラフ線形配置問題である．

$$HLA_\pi := \sum_{x \in X} \left\{ \max_{i \in x} \pi(i) - \min_{j \in x} \pi(j) \right\}$$

3 Median Exchange Iteration

3.1 Median Iteration

MIは，線形配置問題を実数緩和し，解候補の改善を繰り返していくアルゴリズムである．まず，数直線上の整数点に重ならないように配置するところを，実数点・重なって配置可能と緩和する．こうして緩和した問題で，ある頂点v以外の頂点を固定し，頂点vが接続している頂点群の中央値を頂点vの再配置場所とする．これを全ての頂点に対して行い再配置する．この再配置は緩和問題を改善する解である(Koren and Harel,2002)．

MIを次のように拡張することで，ハイパーグラフに対応できる．ある頂点v以外の頂点を固定し，頂点vと接続するすべてのハイパーエッジに対し，頂点vを除いた配置場所の最大値と最小値を取り出し，これらの中央値に再配置する．これにより実数緩和した問題を改善する配置を得られる．ハイパーグラフの場合は，異なるハイパーエッジで同じ頂点と接続していることがあるが，中央値を決める際は別のものとして扱う．

3.2 Median Exchange Iteration

MIは重複配置を許容するが，最終的に前から一つずつ整数点に重ならないように配置し直す．そのため，緩和問題では解が改善していても，MIでの配置と変わる元の問題では，悪化している可能性がある．そこで，本研究ではMIで求めた配置場所を利用して頂点を入れ替えるMedian Exchange Iteration(以下MExIとする)を提案する．図1のように，現在の配置場所と再配置場所がクロスするような2頂点を入れ替えることで，他頂点には影響を与えずに解を改善できる．

Extension of the Median Iteration Algorithm to Hypergraphs
† Takuto SUSUKI / Graduate Scool of Media Arts, Science and Technology, University of Tsukuba
‡ Mikio YAMAMOTO / Institute of Systems and Information Engineering, Unversity of Tsukuba

[1]正確には辺の長さに辺の重みをかけたものの総和を最小化する．本研究では，全ての重みを1とするため省略する．

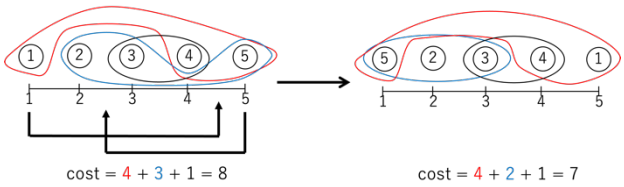

図 1　Median Exchange Iteration

4　実験

提案手法が解の改善に有効か確かめるために実験を行った．実験には，CPU Intel Xeon E5-2620v4 2.10GHz，メインメモリ 256GB の計算機を使用した．実験データはニュースサイト記事を学習した 5gram 言語モデルを使用し，エントリ数(TRIE のノード数)が 500m(5 億)，1G(10 億)の 2 つを用いた．解の初期値はランダムに配置したものを使用し，MI と MExI はそれぞれ 10 回繰り返した．

図 2 と 3 に MI と MExI でのコストの改善の様子を示す．縦軸は TRIE をダブル配列に配置する際の各行の配列幅の平均であり，線形配置問題のコストと対応する．

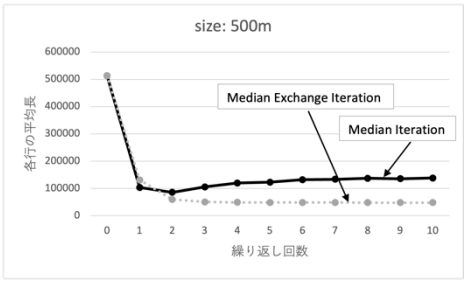

図 2　500m での配列平均長の改善の様子

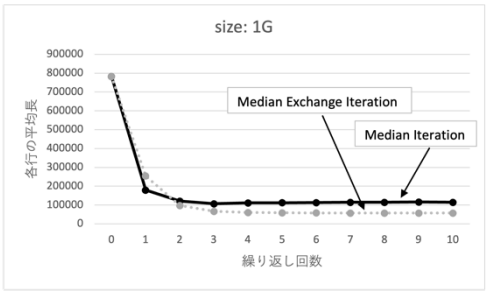

図 3　1G での配列平均長の改善の様子

図 2 と図 3 より，MI と MExI はどちらも解を改善し，MExI の方がより改善していることがわかる．図 2 で，MI は改善した後悪化しているが，MExI は改善し続けている．これは，一つの頂点を移動する際に他の頂点が影響を受けないためだと考えられる．

実行にかかった時間は MI が 500m で 16,863s，1G で 29,746s，MExI が 500m で 40,395s，1G で 101,974s であった．MExI の方が長時間かかるのは，MExI は MI の結果を元に入れ替えるペアを探すためである．また，入れ替える相手を探すことにも時間がかかるため，高速に相手を探すことが課題である．

改善した解(単語 ID)を用いて DALM を構築し充填率を計測した．充填率は 100%に近いほど小さいモデルである．結果は 500m で初期解：47.97%，MI：56.12%，MExI：64.28%，1G で初期解：41.62%，MI：48.19%，MExI：58.86%であった．MI，MExI ともに初期解より充填率が向上している．また，DALM にする前の各行の平均配列長が良い方が充填率も良くなっていることがわかる．

5　おわりに

本研究では，DALM 構築の際の単語 ID 決定の操作をハイパーグラフの線形配置問題とみなして，MI をハイパーグラフに拡張し適用した．このアルゴリズムは緩和問題として解くため，頂点の再配置を移動ではなく入れ替えとすることで，より解を改善できることを示した．また，改善した解(単語 ID)を用いた DALM の充填率が向上することも示した．

今後は，繰り返しの序盤で収束しているため，1 対 1 で入れ替えているところを，n-opt で入れ替えるなど入れ替えの方法を増やすことで近傍を広くし，より解が改善できないか検討する．

参考文献

Aoe, Jun-ichi. An efficient digital search algorithm by using a double-array structure. IEEE Transactions on Software Engineering, Vol.15, No.9, pp. 1066-1077, 1989.

Fredkin, Edward. Trie memory. Communications of the ACM, Vol.3, No.9, pp.490-499, 1960.

Garey, M.R. and D.S. Johnson. Computers and Intractability. A guide to the theory of completeness, W. H. Freeman and Company, New York, 1979.

Jin, R., Y. Xiang, D. Fuhry, and F.F. Dragan. Overlapping matrix pattern visualization: A hypergraph approach. In 2008 Eighth IEEE International Conference on Data Mining, pp.313-322, 2008.

Koren, Yehuda and D. Harel. A multi-scale algorithm for the linear arrangement problem. In International Workshop on Graph-Theoretic Concepts in Computer Science, 2002.

Yasuhara, M., T. Tanaka, J. Norimatsu, and M. Yamamoto. An efficient language model using double-array structures. In Proceedings of the 2013 Conference on EMNLP, pp.222-232.