アルゴリズム 28－4
（1992．7．17）

分枝限定法による系列分割問題の算法構成と効率

加地太一　　大内　東
北海道情報大学　北海道大学

最適系列グラフ分割は系列グラフに対して、各頂点に与えられた重みの総和がブロックサイズ $P(>0)$ 以下であり、かつ、部分集合の頂点番号が連続的に保持される条件のもとで、カットされる辺のコストの和が最小となるよう分割する問題である。著者等はこの系列分割問題の特殊性を有効に利用した分枝限定法による構成を明らかにする。算法の効率性を考察するために、次の6つの構成要素である分枝規則、探索規則、削除規則、優越関係、下限値関数、上限値に着目して算法を構成する。さらに上記の規則のもとで非活性化された節点をクローズド・リストに管理し最良優先探索法を用いることによって、確定性が得られることを明らかにする。以上の確定性が得られる算法の構成法による効率性の関係を示し、さらに $O(nloglogN)$ の計算量が得られることを示す。

Algorithm and Efficiency of Branch-and-Bound for Optimal Sequential Partitions

Taichi KAJI† and Azuma OHUCHI††
†Hokkaido Information University
††Hokkaido University
Sapporo 069, JAPAN

Optimal sequential partitions of graphs is to find a minimum cost partition of the nodes of a graph into subsets of a given size, subjecting to the constraint that the sequence of the nodes may not be changed, that the nodes in the a subset must be of consecutive numbers. We apply the Branch-and-Bound algorithm to the problem. Branch-and-Bound implicit enumeration algorithm has recently emerged as the principal general method for finding optimal solutions for discreat optimization problems. We give an efficient sixtuple characterization for this problem. So, we evaluate the performance of the proposed algorithm, and show that the computing complexity of the algorithm has $O(nloglogN)$time and $O(n)$ space. We present the results of the performance of the algorithm.

1 はじめに

頂点が連続的な番号を保持し、始点が初期番号、終点が最終番号となり、両端点が異なるグラフを系列グラフ $G(V,E)$ と呼ぶ。本論文における最適系列グラフ分割は系列グラフに対して、各頂点に与えられた重みの総和がブロックサイズ $P(>0)$ 以下であり、かつ、部分集合の頂点番号が連続的に保持される条件のもとで、カットされる辺のコストの和が最小となるよう分割する問題である。

この問題に対して Kernighan[1)]の動的計画法（ＤＰ）による研究、また浅野[3)]の区間木によるデータ構造を採用した研究などがある。著者等はこの系列分割問題の特殊性を有効に利用した分枝限定法による構成を明らかにする。分枝限定法はあらゆる組み合わせ最適化問題に適用できる幅広い計算原理であり、実用に耐えるアプローチである。問題の特殊性を最大限利用することによって、上記の研究と同等なあるいはそれ以上の効率性をもち、また問題の複雑性に対処できる応用性の広い算法となる。　算法の効率性を考察するために、次の６つの構成要素である分枝規則、探索規則、削除規則、優越関係、下限値関数、上限値に着目して算法を構成する。さらに上記の規則のもとで非活性化された節点をクローズド・リストに格納し管理することによって確定性が得られることを明らかにする。以上の確定性が得られる算法の構成法による効率性の関係を示し、さらに $O(nloglogN)$ の計算量が得られることを示す。さらに細分化禁止則および確定数から効率性の効果を明らかにする。

2 最適系列分割問題

頂点集合 $v=\{1,2,\ldots,num\}$ 、辺集合 $E=\{(i,j)|1\leq i<j\leq num\}$ からなる無向グラフを $G=\{V,E\}$ とし、頂点数 $num=|V|$ とする。各辺は非負のコスト $C=\{c_{ij}|1\leq i<j\leq num\}$ をもつ。各頂点は $\{w_1,w_2,\ldots,w_{num}\}$ の重みをもち、各々 $0<w_i\leq P$ である。ここで P はブロックサイズと呼ばれる数である。この問題ではすべての重み w_i と P は正の整数とする。また頂点部分集合 $S(\subseteq V)$ の重みは $|S|=\sum_{j\in S}w_j$ とする。G を k 個の部分集合 $\{G_1,G_2,\ldots,G_k\}$ に、以下の制約（ 2.1 ）、（ 2.2 ）のもとで、カットされる辺のコストの総和が最小になるよう分割する：

（ 2.1 ）G_i の頂点の重みの総和　$\leq P$、

（ 2.2 ）任意の G_i の各頂点番号は連続的な番号をもつ。

部分集合 G_i はブロックと呼ばれ $G_i=\{j,j+1,\ldots,m-1,m\}$ とする。G_i での一番小さな頂点番号を b_i として、ブレーク・ポイントと呼ぶ。$b_i=j$ は頂点 $j-1$ と頂点 j の間でカットすることを意味する。集合 $\{b_1,b_2,\ldots,b_k\}$ は一意的に分割 $\{G_1,G_2,\ldots,G_k\}$ を表現する。　ここで、グラフＧが有向グラフの場合、(i,j) と (j,i) の辺を含んでいるのなら、$i<j$ である単一辺 (i,j) におきかえる。その時、コストは $c'_{ij}=c_{ij}+c_{ji}$ とする。さらに便宜上、人為的に $num+1$ の頂点を加え、$c_{num,num+1}=0$ とする。

3 分枝限定法の構成

分枝限定法は最適化問題を分枝と限定をくりかえして解く手法であり、次の6つの構成要素すなわち分枝規則、探索規則、削除規則、優越関係、下限値関数、上限値から成り立つ。ここでは分枝限定法のクローズド・リストを付加したパラダイムにもとずく算法を、系列分割問題に適合させ、その工夫点、考察を各項目にそって述べる。合わせて系列分割問題の対しての分枝限定法の性質について記す。

3．1 6つの構成要素

（1）分枝規則

分枝規則は探索空間を広げていくために選ばれた節点を展開するための規則の集合である。また分枝規則により選ばれた節点の集合を候補者集合という。候補者集合に属する各節点には下に述べる下限値関数により下限値を求め、この値を付与しておく。本問題における分枝規則としては頂点系列の初期値を開始点として次の2つの性質により順次展開していく。

（3.1）各ブロックの頂点の重みの和がPを越えてはならない。

（3.2）最適解において頂点の重さの総和がPとなる任意の頂点の範囲内において、3つ以上のブレイクポイントが置かれることはない。したがっコスト最小化の立場は細分化した状態を無視してよい。これを以後、細分化禁止則と呼ぶ。

以上の2つの性質より、任意の部分解に対して、展開される子部分解の最終ブレーク・ポイントの位置は次の不等式で制約される。ここで$d(x)$はブレーク・ポイントxを始点とし、制約条件（3．1）を満たすブロックの中で最大なものの終点を表ことにする。

$d(BP_i) \geq BP_{i+1} > d(BP_{i-1})$

BP_i：ある任意の部分解πの最終ブレーク・ポイント

BP_{i-1}：部分解πに対する直接の親部分解の最終ブレーク・ポイント

BP_{i+1}：部分解πに対して次に展開される最終ブレーク・ポイント

（2）優越関係

優越関係は部分解の集合で定義された2項関係で、2つの部分解の優劣を比較する。部分解$\pi = \{b_1, b_2, \ldots, b_m\}$と部分解$\pi' = \{b'_1, b'_2, \ldots, b'_n\}$において、$\pi, \pi'$を先祖とする可能解の集合の中でコストが最小なものに対して比較し、2つの部分解の優位性を決定する。πから派生するすべての可能解の中で最小のコスト値がπ'から派生される同様なコスト値より低い値を示すなら、πはπ'に対して優性であり、逆に後者は前者に対して劣性であると言う。劣性のπ'からは最適解は展開されないので削除可能である。優越関係の決定が分枝限定法の効率の決定的要因となる。本問題では優越関係として以下の2つ

を採用する。

（１）同レベルにおける優越関係

（２）細分化における優越関係

同レベルにおける優越関係とは、同レベルな任意の２つの部分解をπ, ρ とした場合、$Cost(\pi) < Cost(\rho)$ ならば、どちらの同レベル部分解π, ρ においても以後派生するブレーク・ポイントのパターンは等しいので、コストの単調増加性よりπはρに対して優性であることが成り立つ。

（３）下限値関数

与えられた部分解π に対して、π を含む全ての可能解の目的関数値の下限値を与える関数である。この関数を $lbd(\pi)$ として表す。現問題では部分解πのコスト値を $lbd(\pi)$ の値として採用している。今後、特に誤解が生じない限り、ある節点が部分解πを指しているとき、その節点自身をπで表す。そこで節点πを節点ρから派生した直接の子節点とすると、πの下限値関数の値は増分コスト関数 $C(x,y)$ を用いて、

$lbd(\pi) = \rho.lbd + C(b_{m+1}, b_m)$

により再帰的に計算できる。ここで、b_mはρの最終ブレーク・ポイントであり、b_{m+1}はρをπに拡大するののに用いたπの最終ブレーク・ポイントである。また節点ρに対して、$\rho.lbd$ はそれまでの探索過程で以前に計算された下限値が存在する場合、その付与された値を示す。　この問題での下限値として部分解のコストをあてる。それを求めるための増分コスト計算は、その定義式である $C(x,y) = \sum_{\substack{y \le i < x \\ j \ge x}} c_{ij}$ を直接用いる方法に対して、以下の漸化式

$$
\begin{aligned}
C(x,y) &= C(x-1,y) \\
&\quad +(c_{x-1,x} + c_{x-1,x+1} + \ldots + c_{x-1,n}) \\
&\quad -(c_{y,x-1} + c_{y+1,x-1} + \ldots + c_{x-2,x-1})
\end{aligned}
$$

を採用し、辺への参照回数を減少させる。またプログラムの実現にあたって、各頂点の流入辺を距離の短い順にリストとしてつなぐことによってスパース部への非参照を行い、流出辺のコストを補助データ表に格納することなどで、さらに参照、および演算回数を減少させ効率化の改善を計る。

（４）上限値

上限値は全ての可能解の目的関数の上限として、現在まで知られている最良値である。

（５）削除規則

削除規則は優越関係や上限値、下限値関数を使って、候補者の削除、現在の活性化節点、および非活性化節点の中から不用のものを除去する規則の集合である。展開された子節点と現オープン・リスト中または現クローズド・リスト中の対象となる要素との優越関係の対比によって、オープン・リスト中またはクローズド・リスト中の対象要素への更新および移動、あるいは子ノードの削除を行う。

（６）選択規則

選択規則は現在の活性化された節点の集合から次の展開対象となる節点を選択する規則の集合である。この規則の設定によって探索経路が決まり、最適解に到達する効率が決定される。本問題では探索過程の現時点でコストの値が最も低い節点を選んで進む最良下限探索戦略を主体として考えるが、他の戦略に応用できるように柔軟にし、他の戦略との比較検討を対応できるようにする。

３．２ 確定性について

説明に入る前にいくつかの用語と記法について述べる。ある時点における任意の部分解のブレーク・ポイントおよび最終ブレーク・ポイントをその部分解に対応する節点のブレーク・ポイントおよび最終ブレーク・ポイントと言う。また部分解 $\{b_1, b_2, \ldots, b_m\}$ にブロック $[b_m, b_{m+1}\}$ を追加し拡張して得られる部分解を

$\{b_1, b_2, \ldots, b_m, b_{m+1}\} = \{b_1, b_2, \ldots, b_m\} + [b_m, b_{m+1})$

と書き表す。同じ最終ブレーク・ポイントをもつ許容部分解の中で最小コスト値をもつものを確定した部分解と言う。以後必要に応じてブレーク・ポイントのことを単にＢＰで表すこともある。

クローズド・リストと最良下限探索法を採用すると、オープン・リストの先頭から取り出されクローズド・リストに格納する節点について次の定理が成立する。

定理１．

クローズド・リストを持ち最良優先探索法を用いたＢＢ算法ではクローズド・リスト中に存在する節点は常に確定した部分解である。

4 算法構成と効率性の関係

本章では分枝限定法の構成においてクローズド・リストを用いる算法を中心に算法構成による動作を詳細化し、効率の関係について考察する。また選択規則で種種の探索戦略を用いることによる算法の性質を明らかにする。まず以下の記号を定義する。

（１）任意の時点におけるオープン・リストを表す集合をΓ、クローズド・リストを表す集合をΛとし、未調査集合をUで表す。

（２）節点$i \in \Gamma$を展開して得られるすべての子節点の集合をC_i、その部分集合である候補者選択集合をD_iで表す。ここで$D_i(\subseteq C_i)$は分枝規則に基ずいて決定される。

（３）オープン・リストのみを使用する算法の構成をＣＳＴ１とし、これにクローズド・

リストを付加したときの構成をＣＳＴ２とする。

分枝限定法で削除規則を適用する場面を、ＣＳＴ２を中心に記述する。任意の候補者節点を$j \in D_i$、またjと同レベルの節点がΓ上またはΛ上に存在するとき、これをj'とする。またΓからj'を削除し、かわりにｊを活性化してΓに加えることをj'をjで更新するということにすれば、

ｂ）ＣＳＴ２の場合、

ｂ．１）$j' \notin \Gamma$かつ$j' \notin \Lambda$なら、$j.lbd = lbd(j)$としてjを活性化し、Γに新規に加える。

ｂ．２）$j' \in \Gamma \cup \Lambda$かつ$lbd(j) \geq j'.lbd$ならｊは破棄する。

ｂ．３）$j' \in \Gamma \cup \Lambda$かつ$lbd(j) < j'.lbd$なら、$(i)j' \in \Gamma$のとき$j'$を$j$で更新する。$(ii)j' \in \Lambda$なら$\Lambda$から$j'$を削除し、$j$を活性化して$\Gamma$に加える。

分枝限定法の算法の効率改善のために考えられる主要な方策は、Γの増大をできるだけ抑制することと、枝刈によるＵの縮小をできるだけ促進することの二つである。この視点から算法構成とその動作を分析する。そのために、まず上記ルールの適用が探索木上の集合Γ, Λ, Uにどの様に作用するかにつにて調べてみる。節点$i \in \Gamma_{old}$を展開することにより、未調査集合Uの中に候補者集合D_iが作り出される。$\Gamma = \Gamma_{old} - \{i\}$として、$D_i$に属する各節点$j$に上記ルールを適用した結果として導かれる動作は次に様に場合分けできる。

（１）.（ｂ．１）の条件部が満たされるとき、一つの新規要素jがUからΓに追加される。

（２）.（ｂ．２）の条件部が満たされるとき、jの破棄にともないjを根とする部分木が探索木から刈り取られる。その結果Uはこの部分木のサイズだけ縮小する。一方Γは不変である。

（３）.（ｂ．３）の条件部が満たされるとき、$(i)j' \in \Gamma$なら、更新にともなってj'を根とする部分木が探索木から刈り取られ、新規要素jがUからΓに追加される。これに対して、$(ii)j' \in \Lambda$のときj'自身はΛから削除されるが枝刈は起こらない。そして、新規要素jはUからΓに追加される。

またD_iを生成する段階で、分枝規則で用いられている細分化禁止則により$C_i - D_i$をルートとする部分木も削除されることを注意しておく。これらから、結果はすべて”新規要素をΓに追加する”か”枝刈によってUを縮小する”という二つの動作に帰着される。またＣＳＴ２はＣＳＴ１よりΓの増大の抑制する効果とUの縮小を促進する効果は大であることが生起する条件の包含関係から容易に理解される。

またＣＳＴ２は汎用性に富んだ構成であり、選択規則の戦略を変えることによって、種種の最適化論理構成をなす。縦型探索戦略、横型探索戦略、あるいはヒューリスティク関数の導入で、あらゆる問題適合型になりうる。

ここで、レベル値による整列性を保持する戦略を用いることによって、算法の第$(m+1)$段の開始直前のクローズド・リストとオープン・リストの状態とそのオープン・リストの要素の構成は以下のように成立することが示せる。

クローズド・リスト：$(i_1^{(1)}, \ldots, i_{m-p+1}^{(m-p+1)}, \ldots, i_{m-1}^{(m-1)})$

オープン・リスト：$(i_m^{(m)}, i_{m+1}^{(m)}, \ldots, i_{m+p-1}^{(m)})$

$$i_m^{(m)} = min\{j_m^{(m-p)}, j_m^{(m-p+1)}, \ldots, j_m^{(m-1)}\}_{lbd}$$
$$i_{m+1}^{(m)} = min\{j_{m+1}^{(m-p+1)}, \ldots, j_{m+1}^{(m-1)}\}_{lbd}$$
$$\cdots$$
$$i_{m+p-1}^{(m)} = min\{j_{m+p-1}^{(m-1)}\}_{lbd}$$

ここで $min\{\ldots\}_{lbd}$ は下限値の最小となる節点を選び、その各要素は以下の式で表せる。$j_k^{(m)} = i_m^{(m)}.lbd + c_{m,k}$ また上付の添字は算法の段数を、下付の添字はその節点のレベル数を表している。オープン・リストの要素 $i_m^{(m)}$ は算法の第 m 段であり、次の展開の対象となりクローズド・リストへ確定され移動する節点である。ここで、$i_m^{(m)}$ の確定計算法は以上の漸化式となるが、これは Kernighan のＤＰアルゴリズムと一致することが示される。

次に、最良下限探索戦略を用い前出の定理１の条件が満たされるときΛの節点はすべて確定した部分解に対応する。このような確定性が利用できるときに、ＣＳＴ２において $j' \in \Lambda$ のときの２つの処理が次の単一の規則に縮約される。

$j' \in \Lambda$ のとき j は捨てる

これより確定性が利用できる場合と出来ない場合の動作と効率への寄与を論ずる。ある節点が非活性化されΛに加えられたとき、前者ではこの節点と同レベルにある U 中のすべての節点について潜在的に枝刈が行われ U が縮小されることになるが、後者では U 中の同レベル節点の中である節点の lbd 値がより改善されていることが見いだされたとき、その節点のΓへの追加と元の節点のΛからの削除が行われる。すなわち前者では非活性化が生ずる度に確定される節点のレベル値はすべて異なるが、後者ではあるレベル値の節点が確定するまでに何回かの非活性化の過程を経る可能性がある。それゆえ、前者では最終レベルの節点すなわち最適解を確定するのに要する非活性化の回数が高々 num であるのに対して、後者では少なくともこの回数を越える非活性化が必要になる。

以上、クローズド・リストを付加し、最良優先探索を用いることによって効率的算法が得られ、以下に示す精密な計算量が得られた。第一に算法全体でのコスト計算に関する複雑性に着目する。その計算量は辺への参照回数と演算数により構成され、コスト計算の漸化式とデータ構造の工夫を考慮することによってコストに関する総計算量は $O($辺数＋頂点数×ブロックサイズ$)$ となり、スパースなグラフでは $O(n)$ となる。第２に増分コストを除く全体の計算量はヒープに対する操作となるので、ヒープから最小値を取り出す操作、これは定理から高々 n 回となり、ヒープの要素の値を書き換える操作が最大で全分枝数の回数行われ、ヒープに対する処理が $O(logn)$ となるので、以上の操作は $O((頂点数＋全分枝数)\ logn) \approx O(nlogn)$ となる。したがって、前者と後者を合わせて分枝限定法の計算量は $O(nlogn)$ と考えられる。

さらに優先待ち行列に Van Emde-Boas priority queues[6] を用いることによって、最小値の取り出し、要素の値の書換操作が $O(loglogN)$ で行える。ここで N は下限値集合の大きさである。これより、全体の計算量は $O(nloglogN)$ となり、ある限定範囲で有為な計算量が得られる。

5 おわりに

分枝限定法の算法の効率改善のために考えられる主要な方策は活性化された節点を格納するオープン・リストの増大をできるだけ抑制することと、枝刈による未調査な節点の集合の縮小をできるだけ促進することの二つである。この視点から6つの構成要素である分枝規則、選択規則、削除規則、優越関係、下限値関数、上限値の構築を行ない、最良下限探索法のもとでクローズド・リストを用意することによって確定性の性質が得られ、高効率な算法が実現できる。

また分枝規則における細分化禁止則の子ノードの枝刈によるUの縮小の効果と選択規則で最良下限探索法を用いることによっての確定数の減少によるΓの増大抑制の効果を数値的に調べてみると、細分化禁止則によって約20%から45%の削除効果があげられ、確定数はノード数以下の値となる。また細分化禁止則による効果と確定数の減少効果もブロックサイズが増加するとともに上がる傾向にある。

参考文献

1)Brian.W.Kernighan: Optimal Sequential Partitions of Graphs, J.ACM,Vol.18,No.1. pp.34 -40,1971.

2)Walter.H.Kohler: Characterization and Theoretical Comparison of Branch-and-Bound Algorithms for Permutation, J.ACM,Vol.21,No.1. pp.140-156,1974.

3)T.Asano: Dynamic Programing on Intervals, Technical Report 91-AL-20, Information Processing Society of Japan,pp.1-8, 1991.

4)T.Kaji and A.Ohuchi: Optimal Sequntial Partitions of Graphs, Technical Report 90-AL-10, Information Processing Society of Japan, pp.1-8, 1991.

5)T.Kaji and A.Ohuchi: Complexity of Optimal Sequential Partitions of Graphs by Dynamic Programming, Bulletin of the Faculty of Enginering Hokkaido University, No.157,pp.59-70, 1991.

6)P.van Emde Boas,R.Kaas, and E.Zijlstra: Design and Implementation of an Efficient Priority Queue, Mathmatical systems theory,10, pp.99-127,1977.