分枝限定法における特性と性能評価*

3X−1

加地太一[†] 大内 東[‡]

北海道情報大学 北海道大学

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．
著者等はこの系列分割問題の特殊性を有効に利用し、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築した．本稿では分枝限定法の構成にあたって考慮した諸要素が算法特性と性能におよぼす影響を理論的ならびに実験的に調べ、各種の諸要素と戦略の選択による算法の諸特性と性能の比較について述べる．その検討の結果、有効なものとしてレベル順横型探索と最良下界探索が取り上げられ、さらに詳細な検討を試みた．

2 汎用型プログラムの実装

以下の７個の項目の全てを実装するプログラムを作成し、これを汎用型プログラムと呼ぶ．

- 細分化禁止則による候補者集合の絞り込み
- クローズド・リストの採用
- オープン・リスト（Γ）、クローズド・リスト（Λ）によるマーク機能とマーク要素の後処理
- 新規要素の上限値による削除機能
- 上限値の自由変更とＵカット機能
- 戦略選択が可能である
- 汎用停止則の採用

3 算法構成と効率性の関係

本章では前章の分枝限定法の構成にあたって考慮した諸要因が算法特性と性能におよぼす影響を、以下の評価項目に着目して理論的ならびに実験的に調べ評価を行う．

- 細分化禁止則による候補者の絞り込み
- 優先待ち行列の長さ
- 確定性の存在
- 停止にいたるまでの展開数
- 上限値による削除効率
- 優越関係による削除効率
- 探索のために利用する機能数
- 特殊化したときのプログラム化の容易さ

まずクローズド・リストの導入による効果を分析し、本問題におけるその取り込みが非常に有効であることについて述べる．次に優先待ち行列におけるキー値の構成法によって各種の探索戦略が導かれ、従来個別の算法として認識されていたものが同一の枠組みの中で実現されることを示すとともに、この立場から各種戦略の選択による算法の諸特性と性能の比較を試みる．さらに細分化禁止則がどのような戦略選択に対しても共通に適用可能であり、ほぼ同様の効果が得られることを実験的に実証する．最後に算法の計算量と優先待ち行列の構成との関係について述べる．

3.1 クローズド・リスト導入による効果

本来、分枝限定法プログラムではクローズド・リストを利用しなくてもオープン・リストのみで構成できるが、本問題ではクローズド・リストの採用によって２つの利点が得られることを示す．まずクローズド・リストの導入によって以後に述べる確定性を有効に利用できるようになる．さらに、クローズド・リストを用いたプログラムはオープン・リストのみを利用したプログラムに比べて，クローズド・リストによりマークされた要素の破棄と新規要素の判断基準が強くなることが導かれる．

3.2 探索戦略による各種算法の実現

汎用型プログラムは選択規則の選択が自由に行えるよう汎用性のある構成で実装されており、縦型探索、横型探索、ヒューリスティクな探索戦略など種々の問題に柔軟に適合できる．この算法において優先待ち行列のキー値を変更することによって同じ枠組みの中で容易に個別の算法が実現される．キー値をノード数−最終ブレーク・ポイント＋１とすると縦型探索戦略、最終ブレーク・ポイントとするとレベル順横型戦略、下限値とすると最良下界探索、およびヒューリスティック関数を用いると経験則にもとずいた探索となる．ここではレベル順横型戦略と最良下界探索について述べる．

3.2.1 レベル順横型探索

優先待ち行列のキー値を最終ブレーク・ポイントの値に設定することによって、オープン・リスト中の要素はブロックサイズ分、レベル順に整列された集合となり、探索木を幅優先で進こととなるので、この戦略をレベル順横型戦略と呼ぶ．この戦略を用いた場合、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・リストの内容を以下に詳細に示す．
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：$(i_m^{(m)},i_{m+1}^{(m)},\ldots,i_{m+p-1}^{(m)})$

---
*Property and Ability for Branch-and-Bound
†Taich Kaji, Hokkaido Information University
‡Azuma Ohuchi, Hokkaido University

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

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[1)]のＤＰアルゴリズムと一致することが示される．

ここで算法の特徴を述べる．まず細分化禁止則による絞り込みは自動的に組み込まれる．オープン・リスト長は定常的にはブロックサイズと一致し、Ｕｃｕｔが起きた時点で過渡的な減少を示す．汎用停止則を使用のもとではＵｃｕｔの作用により、ｎ＋１段より早い停止が起こる．それに対して、Ｕｃｕｔを用いず、さらにｎ＋１回の反復で停止させることにしても最適解は求まるが、ステップ数の増大が起こる．これから細分化禁止則の適用をはずしたものがＤＰと完全に一致している．レベル順横型戦略に対応する動的計画法では優先待ち行列を用いない構成をとっているが、この場合、細分化禁止則とＵｃｕｔなどの導入が難しくなる．

3.2.2 最良下界探索

クローズド・リストと最良下界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．

この算法の特徴について述べる．系列分割問題における固有の分枝規則として導入した細分化禁止則は当然これにも用いられており、その効果については後にまとめて述べる．オープン・リスト長は段数ごとに変化するが、平均長はブロックサイズ長をやや上まわる程度であることが判明している．節点は下限値の大きさの順に一つずつ確定されていく．それゆえ、多くともｎ＋１回の反復ですべての節点の確定が行われ、実際にレベル値ｎ＋１の節点はそれ以前に確定される．それでレベル値ｎ＋１に達したことを停止則にとれば汎用停止則より有利になる．この時、停止するまでの段数は、実験的に確かめた結果、約７０％前後の数であった．この数はブロックサイズの増大とともに減少する傾向がある．特殊化された停止則を使うことにすれば、上限値による削除は不必要になる．優越関係による削除効率はレベル順横型探索に対してやや上回った数値を得た．確定性を用いると、上限値による削除規則と、Λから Γへの更新が不必要になる．従って削除規則に含まれる判断肢の簡略化、削除が可能となりその分オーバーヘッドを減少させることができる．

3.3 細分化禁止則導入による効果

節点ｉ∈Γを展開して得られるすべての子節点の集合を c_i、それから細分化禁止則にもとずいて決定される候補者集合を d_iとすると、d_iを生成する段階で、$c_i - d_i$の各節点をルートとする全ての部分木が削除され探索効率が改善される．この細分化禁止則による候補者の絞り込みが本問題の算法構成に取って有効であることを数値実験により確かめた．その結果、細分化禁止則によってどの探索戦略に対しても約２０％から４５％の削除効果が得られ、この効果はブロックサイズ増加ともに向上することが示された．

3.4 待ち行列と計算量

計算量に寄与する主たる要因は増分コスト計算ならびに待ち行列の基本操作に要する計算量と停止に至るまでに待ち行列に対して行われる基本操作の総数との積である．分枝限定法の計算量は $O(nlogn)$ と考えられる．レベル順横型探索ではレベル値をキーとして優先待ち行列を構成していることに着目すれば、単純待ち行列の使用によってより効率の良い算法構成が可能である．たとえばレベル値を添え字とする大きさｎの一次元配列上を移動する待ち行列を用いることによって、基本操作は全て添え字による一動作で行うことが可能となり、もっともオーバーヘッドの少ない算法構成ができる．これから細分化禁止則とＵｃｕｔ機能を取り除けばＤＰの実装そのものになる．この意味でこの算法によってＤＰの改良版が得られる．また最良下界探索の理論的計算量を改善する算法を優先待ち行列に Van Emde-Boas priority queues[5)]を用いることによって実現できる．この待ち行列では最小値の取り出し、要素の値の挿入、書換操作が $O(loglogN)$ で行える．ここでＮはキー値の上界である．Ｎとして下限値集合の上界を取り、現問題でコスト値が有界である場合を考えるとＮはｎに比例すると考えて良いので計算量は $O(nloglogn)$ になる．

4 おわりに

分枝限定法の枠組みの中で動的計画法を含む種々の算法構成を実現し、比較検討を行ない、ステップ数に関しては最良下界探索法による優位性を示され、待ち行列長ではレベル順横型探索の優位性が示された．細分化禁止則やレベルによる削除率などは同程度の効果を得た．Ｐを固定する理論的計算量はレベル順横型探索では $O(n)$ であり、最良下界探索では $O(nlogn)$ となり、優先待ち行列の工夫によって $O(nloglogn)$ まで下げられることを示した．しかし、待ち行列長に関する経験則からこの計算量は $O(n)$ に近いものと見られる．さらに細分化禁止則、ステップ数減少、レベルによる削除などの探索空間のしぼりこみによって複雑性の負担が軽減され、レベル順横型探索をもとにして、これらの諸特性を備えたＤＰの改良版を得ている．

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: Algorithm and Efficiency of Branch-and-Bound for Optimal Sequential Partitions,Technical Report 92-AL-28, Information Processing Society of Japan,pp.25-32, 1992.
5)P.van Emde Boas,R.Kaas, and E.Zijlstra: Design and Implementation of an Efficient Priority Queue, Mathematical systems theory,10, pp.99-127,1977.