ラグランジュ緩和を用いた最小根付きk-部分木問題の最適解法

02004490　防衛大学校情報工学科　荒木紀雄　ARAKI Norio
01107880　防衛大学校情報工学科　*片岡靖詞　KATAOKA Seiji

1　はじめに

点集合 V および枝集合 E からなる無向連結グラフ $G=(V,E)$ に対して，根となる点 r と整数 k が与えられたとき，r を根として枝の数が丁度 k 本であるような連結部分木を**根付きk-部分木**と呼ぶことにする．本研究では，各枝に重みが与えられているとき，枝の重みの総和が最小になる根付き k-部分木を求める問題を扱い，**最小根付きk-部分木問題** (Minimum k-Subtree Problem : k-STP) と呼ぶことにする．本発表ではこれまで研究・提案してきた上下界値算法 [1, 2] やラグランジュ緩和法による下界値の改善結果 [3] に基づいて，分枝限定法アルゴリズムを開発し，その計算機実験結果を報告する．

2　これまでの研究 [1, 2, 3, 4]

根 r からグラフを幅優先探索した探索木における枝のレベルを枝の**歩数**と呼ぶ．このとき，閉路を生成しないように，i 本目の枝を歩数 $1,2,\ldots,i$ $(i=1,2,\ldots,k)$ の枝集合から選ぶ枝集合族を $\boldsymbol{X}$ とする．このとき，各繰り返しにおいて最小の重みを持つ枝を選べば，その解は k-STP の下界値を与えることを示した．この方法を **GLB**(Greedy Lower Bound) と呼ぶ．

また，枝集合 $\boldsymbol{x}$ を (1)$|\boldsymbol{x}|=k$; (2) 閉路を構成しない ; (3)$\boldsymbol{x}$ の枝に対し i 番目の枝の歩数が i 以下になるような番号 i $(i=1,2,\ldots,k)$ をつけることができる ; の3条件を満足する集合とすると，その集合族は $\boldsymbol{X}$ と一致することを示した．このとき，k-STP は次のように定式化できる．ただし，w_{ij} は枝 (i,j) の重み，S を $r\in S$ かつ $|S|\leq k$ である点集合，$\overline{S}$ を S の補集合とする．

$$
(\mathbf{P})\quad
\begin{array}{lll}
\min & \sum_{i\in V}\sum_{j\in V} w_{ij}x_{ij} & (1)\\
\text{s.t.} & \boldsymbol{x}\in \boldsymbol{X} & (2)\\
 & \sum_{i\in S}\sum_{j\in \overline{S}} x_{ij}\geq 1 \quad \forall S & (3)
\end{array}
$$

これは，(P) から連結性を保持する条件 (3) 式を除去した緩和問題は GLB により最適に解けることを意味している．しかも，この解は整数解になっている．これらの利点を活かし，さらに下界値を改善するために (3) 式をラグランジュ緩和することを考えた．

(3) 式は膨大な数があるため，そのすべてを目的関数に組み込むことは難しい．そこで，GLB を用いて緩和問題を解き，得られた解で連結性を破っている制約を切除平面法のように逐次加えながら同時にラグランジュ乗数を更新する方法を提案した．組み込む制約は不等式制約であるため，実行可能性と相補性の両方が満足されないと最適である保証は得られないが，劣勾配法を巧みに用いることにより，従来の下界値よりも 5 ～ 10%の改善を達成することができた．さらに計算の途中で上界値も改善されるという利点も得られた．

3　k-STP の分枝限定法

定式化 (P) では，決定変数が枝に対応しているため，枝を分枝変数とするのが常識的である．しかし，k-STP の特徴として，k-STP の解は，その解でカバーされている点集合での最小木に一致する性質を持つ．したがって，$k+1$ 個の点を最適に選ぶことができれば，k-STP の解を得ることができる．したがって，点を分枝変数とする分枝限定法も考えることが可能である．

また，枝 (点) を解に含むように固定する場合，必ず根からの連結性を保つように固定しなければならない．そうでないと，スタイナー木問題になってしまうためこれまでの議論が適用できなくなることに注意する．以上のことに留意すれば，解に含むように固定された枝 (点) は，**拡大された１つの根**ととらえることができる．このとき，多重枝が生成されることがあるが，上記の k-STP の解とカバーされた点における最小木が一致するという性質から，重みの最も小さな枝のみを残せばよいことがわかる．

3.1　枝を分枝変数とする場合

枝を分枝変数とする場合，分枝の一方では最適解が得られやすく，他方では子問題の下解値ができるだけ上昇するように選択するのが望ましい．本研究では親問題に GLB を適用したときに，1 番目に選ばれる枝に対応する変数を分枝変数とする．また，分枝頂点の選択方法としては，分枝変数の枝を採る側の子問題を先に解く奥行き優先とする．分枝の停止条件は (1) 実行可能解が求

められた場合，(2) 下界値が上界値を超えた場合に加えて，(3) 選択しない枝の両端点が，選択しなければならない点となる場合の 3 条件とする．(3) の条件で分枝を停止できる理由を示す．図 1 において，分枝変数として

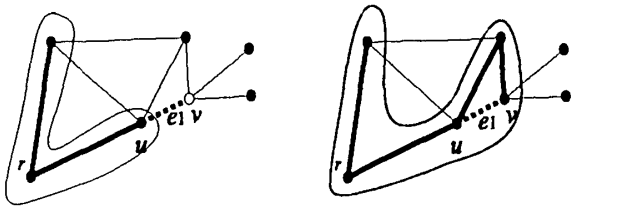

図 1: 親問題 (左) と分枝を停止してよい子問題 (右)

選ばれた枝 $e_1(u,v)$ を採らない子問題を生成した結果，点 v と拡大された根が e_1 以外の枝を介して連結した子問題が生成されたとする．しかし，この子問題からは最適解は求められない．なぜなら，点 v を含む新たな拡大された根の中は最小木でなければならないが，この拡大された根の中で e_1 以外の枝を用いても最小木が得られないからである．

3.2 点を分枝変数とする場合

点を分枝変数とするときも，枝を分枝変数とする場合と本質的には変わらない．したがって，親問題に GLB を適用したときに 1 番目に選ばれた枝において，拡大された根でない方の点を分枝変数として選択する．前節で述べた停止条件 (3) と同じ理由から，点を採らない側に固定した場合，その点に接続するすべての枝を除去することができるため，問題を縮小することができる．また，分枝頂点の選択方法としては，分枝変数となる点を採る側の子問題を先に解く奥行き優先とする．分枝の停止の条件は (1) 実行可能解が求められた場合，(2) 下界値が上界値を超えた場合の 2 条件とする．

4 計算機実験

本研究で提案する手法に対し，C 言語でプログラムを作成し計算機実験を行う．計算機は OKITAC9000 を使用した．枝の重みを 2 点間のユークリッド距離とし，この値の小さい枝から順に選択したグラフを使用した．ラグランジュ乗数更新のための繰り返し回数を 5 回 (L05)，10 回 (L10)，20 回 (L20) とし，下界値として GLB(繰り返し 0 回に対応．[2] で採用) を使用した分枝限定法と，生成された子問題数及び実行時間を比較した．また，劣勾配法における各種パラメータは [3] に従った．表 1 及び 2 に，下界値の違いによる結果を，枝を分枝

表 1: 下界値の違いによる結果 (分枝変数：枝)

$\lvert V\rvert$	$\lvert E\rvert$	k	GLB	L05	L10	L20
20	47	10	496	216	109	92
			0.2	0.2	0.2	0.3
30	108	15	11090	2174	1624	1443
			8.3	4.4	5.4	9.8
40	195	20	314213	100710	62868	47412
			1398.7	317.3	353.1	527.8
50	306	25	500000	500000	500000	500000
			—	—	—	—

表 2: 下界値の違いによる結果 (分枝変数：点)

$\lvert V\rvert$	$\lvert E\rvert$	k	GLB	L05	L10	L20
20	47	10	208	86	46	22
			0.1	0.1	0.1	0.1
30	108	15	2104	508	388	246
			1.6	1.0	1.4	1.7
40	195	20	109024	30116	15554	2892
			144.8	101.6	91.4	34.8
50	306	25	500000	194832	143580	45304
			—	1014.1	1399.9	967.6

変数とした場合と点を分枝変数にした場合について示した．また，表の上段には生成された子問題数，下段には実行時間 (CPU 時間：秒) を示している．実行中に子問題数が 500000 を超えた場合，最適解が得られていなくても終了させている．

表 1 及び 2 から，下界値として本研究で提案する方法を採用したことにより，子問題数，実行時間とも減少しており，本研究で提案する下界値算法が効果的であることが分かる．また，本研究の方法による下界値は，繰り返しの初期に急激に改善されるので [3]，5 回程度の繰り返しで充分な効果を発揮することが分かる．また，表 1 及び 2 において同じ下界値を用いた場合を比較すると，分枝変数を点にすることで問題が縮小され，実行時間がより短縮されることが分かる．本研究の方法で得られた下界値を組合せて，更に良い効果を得ることができた．

参考文献

[1] 片岡ら，1995 年日本 OR 学会春季研究発表会,2-D-9.

[2] 星崎ら，1996 年日本 OR 学会春季研究発表会,1-B-9.

[3] 荒木ら，1998 年日本 OR 学会秋季研究発表会,1-C-6.

[4] 荒木，防衛大学校理工学研究科, 修士論文,1999.