2000年度日本オペレーションズ・リサーチ学会
春季研究発表会

コストαの全域木を検出するアルゴリズム

02302540　防衛大学校情報工学科　*高橋 元法　TAKAHASHI Motonori
01700900　防衛大学校情報工学科　山田 武夫　YAMADA Takeo

1　はじめに

全域木 [1] はグラフ理論における基本概念で, これに関する問題としては最小全域木問題, 全域木の全列挙 [2], コストが最小から k 番目までの全域木を求める k −全域木問題 [3] などが今までに研究されてきた. 本稿では V, E をそれぞれ節点, 枝の集合とする連結な無向グラフ $G=(V,E)$ と, 各枝のコスト $c: E \to Z^+$ が与えられた場合に, コストが丁度 α となる全域木を求める問題を考察する.

2　分割統治法

G における**最小全域木**は Kruskal, Prim などのアルゴリズムにより効率的に求められるが, このときのコストを $\underline{z}(G)$ と書く. コストの符号を逆転して考えれば, **最大全域木**も同様に求められ, コスト $\overline{z}(G)$ が得られる. $\underline{z}(G), \overline{z}(G)$ は G における全域木の下界値と上界値で,

$$\underline{z}(G) \le \alpha \le \overline{z}(G) \qquad (1)$$

でない場合にはコスト α の全域木は存在しない.

(1) 式が満足される場合には以下のように**分割統治** [4] の考え方を用いる. まず G の全域木 T を一つ求める. これは上述の最大（最小）全域木でも, それ以外でも良い. $c(T)=\alpha$ の場合は問題は解けているので, 以下では $c(T) \ne \alpha$ とする. T の枝の集合を $\{e^1, e^2, \cdots, e^u\}$ $(u=|V|-1)$ とし, **部分問題** P^i を次のように設定する.

P^i : $F^i := \{e^1, e^2, \cdots, e^{i-1}\}$ を含み, $R^i := \{e^i\}$ を含まない G の全域木で, コスト α のものを求めよ.

明らかに, $P^1, P^2, \cdots, P^u$ を解けば元の問題が解けたことになる.

さらに一般化して, F, R を G の枝の部分集合で, 互いに素なものとする. このとき, F をすべて含み, R は含まないような全域木は (F,R)-**許容**であるといい, (F,R)-許容な全域木でコストが α のものを求める問題を部分問題 $P(F,R)$ と表記すると, 元問題は $P(\emptyset,\emptyset)$ となる.

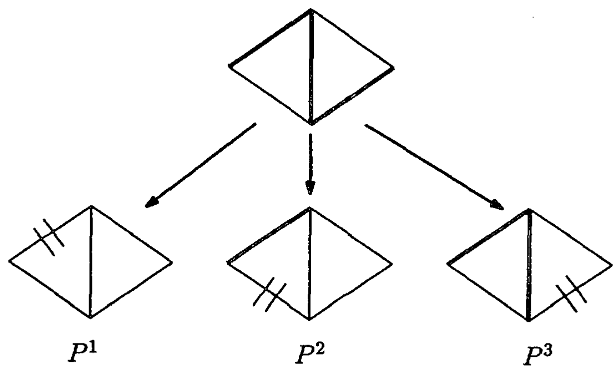

図 1: 全域木と部分問題

部分問題においても, (F,R)-許容な最大および最小全域木は Kruskal や Prim のアルゴリズムを多少修正して容易に求められる. これらのコストを $\underline{z}(F,R), \overline{z}(F,R)$ とすると,

$$\underline{z}(F,R) \le \alpha \le \overline{z}(F,R) \qquad (2)$$

を満たさない場合には部分問題 $P(F,R)$ にはコスト α の全域木は存在しない. (2) が満足される場合には (F,R)-許容な全域木を用いて $P(F,R)$ をさらに**子問題**に分割する.

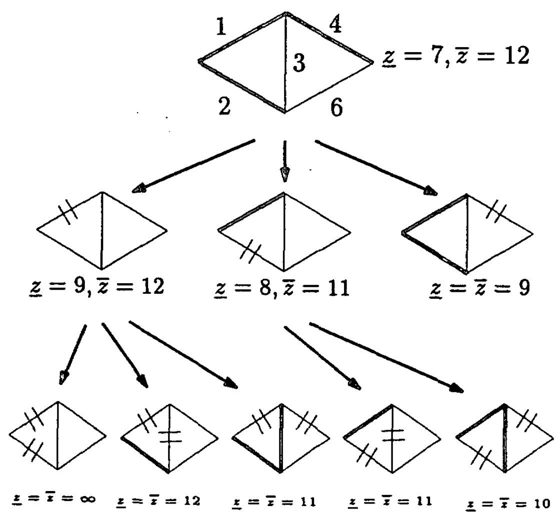

図 2: 分割統治法の挙動

3 数値例

図 2 に, 上のアルゴリズムの適用例を示す. $\alpha=10$ の場合, 最下段の右端でコスト 10 の全域木が見つかってアルゴリズムが終了している.

もう少し大きい例題として, 図 3 のグラフ USA を用いる. この場合, $\underline{z}=1718$, $\overline{z}=3561$ で, $\alpha=3500$ とすると 2451 個の部分問題を生成して図中の全域木が得られる. また, 例えば $\alpha=3536$ の全域木は存在しないことが確認される.

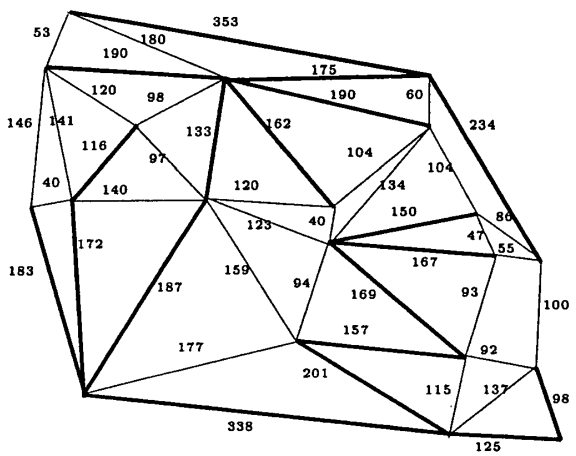

図 3: 例題 USA

4 アルゴリズムの改善

上の方法の改善として, 2 点あげる. まず, 部分問題を子問題に分割していくときに用いる全域木であるが, 上述の数値例では常に最小全域木を用いた. しかし, 例えば $\alpha \in [\underline{z}, (\underline{z}+\overline{z})/2]$ の場合は最小全域木を用い, $\alpha \in [(\underline{z}+\overline{z})/2, \overline{z}]$ では最大全域木を用いる, という方式も可能である. USA で $\alpha=3500$ の例ではこれにより生成部分問題数が 223 個に減少した. α が $\underline{z}$ 付近であった場合と $\overline{z}$ に近い場合についての生成問題数の比較を表 1 に示す。表 1 より, この改善は大きい α に対して効果があることが分かる.

もう一点は, 部分問題 $P(F,R)$ 中にコスト α の全域木が存在しない場合に, 積極的にそのことを検出することを試みることである. すなわち, 全域木という条件を緩和し, 単に $|V|-1$ 本の枝集合と考えると, 上の部分問題では $E\backslash F\backslash R$ 中にコスト $\alpha - c(F)$ の $|V|-|F|-1$ 本の枝集合が存在するか否かを問う問題となる. これは動的計画法で容易に判定出来, そのような枝集合が存在しなければ $P(F,R)$ は終端されるし, この枝集合がたまたま木であった場合にはコスト α の木が見つかってアルゴリズムは終了する.

表 1: 生成部分問題数の比較

α	改善なし	改善あり	α	改善なし	改善あり
1718	1	1	3550	1937	128
1719	20	20	3551	278	4
1720	20	20	3552	5	5
1721	9	9	3553	1153	74
1722	33	33	3554	1153	74
1723	6	6	3555	1153	74
1724	49	49	3556	3	2
1725	45	45	3557	4	5
1726	9	9	3558	320	20
1727	72	72	3559	320	20
1728	47	47	3560	320	20
1729	24	24	3561	1	1

5 結び

本アルゴリズムを一部修正することにより以下の問題についても解くことが出来る.

- コスト α の全域木の全列挙
- コスト α 以下の全域木の全列挙
- コストが $[\alpha, \beta]$ 間の全域木の検出および全列挙

また, 4 で述べたアルゴリズムの改善法のうち, 第 2 点についての数値実験は, 当日報告する.

参考文献

[1] R.K. Ahuja, *et al.*, *Network Flows*, Prentice-Hall, 1993.

[2] Y. Matsui, *et al.*, Algorithm for Combinatorial Enumeration Problem, `http://dmawww.epfl.ch/roso.mosaic/kf/enum/comb/combenum.html`

[3] H.W. Hamacher, *et al.*, "k best solutions to combinatorial optimization problems", *Annals of Operations Research*, Vol.4(1985)

[4] T.H. Cormen, *et al.*, *Introduction to Algorithms*, McGraw-Hill, 1990.