パリティ全域木問題の一解法*

01700900　防衛大学校情報工学科　山田 武夫† YAMADA Takeo

1　はじめに

無向グラフ $G=(V,E)$ において、枝数 $m:=|E|$ が偶数で、枝集合は $m/2$ 個の対 E_i に分割されているとする（図1）．すなわち、$E_i \subseteq E, |E_i|=2\ (i=1,\ldots,m/2)$. G の全域木 T で、E_i が奇パリティ対であるとは、E_i の枝の一方が T に含まれ、他方はそうでないことをいう．奇パリティ対を含まない全域木をパリティ全域木という．すなわち、パリティ全域木では各枝対の2本の枝は同時に T に含まれるか、含まれないかである．全域木の枝数は $n-1$ なので、パリティ全域木が存在するためには n は奇数でなければならない．G にパリティ全域木が存在するかどうかを判定する問題を全域木パリティ問題と言うが、これが $\mathcal{NP}$-完全か否かは未解決 [1] である．

本稿の目的は、G 中にパリティ全域木が存在するかどうかを判定し、存在する場合にはそのような全域木の一つを実際に求めるアルゴリズムを提示する事である．これを以下ではパリティ全域木問題と呼ぶ．

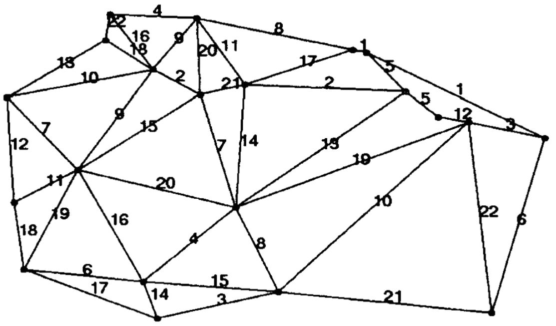

図 1: 例題

2　近似解法

全域木 T に含まれる奇パリティ対の数を $z(T)$ と記し、T の評価値という．問題は評価値 0 の全域木 T^* を求めることであるが、本節ではもう少し要求を緩めて、“出来るだけ小さい評価値を持つ全域木を求めよ”、という問題を考える．このために、最初パリティ条件を全く無視して、Kruskal の方法（例えば [2]）などにより適当な全域木を求め、これを T とする．次に、以下のようにして評価値を逐次改善する．

T において、奇パリティ対が存在しなければ問題はすでに解決しているので、T には奇パリティ対が存在するとする．そのような対の一つに含まれ、T には含まれない枝を e_{in} とすると、$\{e_{in}\}\cup T$ には単純サイクルが一意的に含まれる．この単純サイクル上の e_{in} 以外の枝で、奇パリティ対に属するものがあればこれを e_{out} とする．以上のような e_{in}、e_{out} の組が見つかれば、新たに全域木 $T':=T\cup\{e_{in}\}\setminus\{e_{out}\}$ を得るが、この場合

$$z(T')=z(T)-2$$

で、全域木は確かに改善されている．上のような枝の組が見つからない場合は、その時点での全域木を近似解として出力し、アリゴリズムは終了する．

例 1 図1では Kruskal 法により評価値 12 の初期解を得た．それに上の改善を5回反復して図2の近似解が得られた．ここに破線は奇パリティ対を示し，このときの評価値は2である．

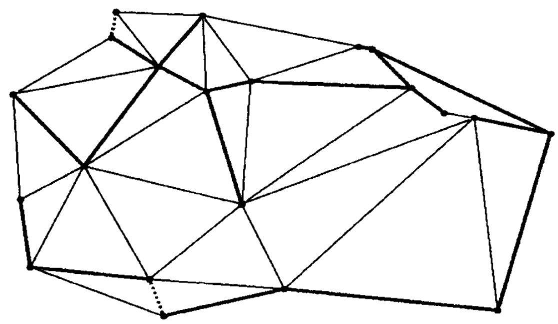

図 2: 近似解

3　厳密解法

本節では分割統治[3] の考え方を用いて、問題の厳密解法を導く．

全域木 T が奇パリティ対を含む場合、そのような対の一つを用いて問題を次のように分割する．ひとつはその対の枝を両方とも含むパリティ全域木を求める問題で、他方は枝をどちらも含まないパリティ全域木を求

*岡山理科大学, 2001.9.12-13
†E-mail: yamada@nda.ac.jp

める問題である．前者では，当該パリティ対が固定されたといい，後者では禁止されたという．一般に，F，R をそれぞれ固定，または禁止された対の集合とし，その条件下でのパリティ全域木問題を **Parity_ST**(F,R) と記す．Fに含まれるすべての枝を必ず含み，Rの枝は全く含まないような全域木は(F,R)-許容であるというと，**Parity_ST**(F,R) は (F,R)-許容なパリティ全域木を求める問題である．

ところで，前節の近似解法を (F,R)-許容という条件付きの場合に拡張することは容易であるが，これを **Find_Approx_Parity_ST**(F,R) とする．これは，(F,R)-許容な全域木が存在する場合には，(F,R)-許容な近似解を返し，そうでない場合には空集合を返す．すると，問題 **Parity_ST**(F,R) を解くアルゴリズム **Find_Parity_ST**(F,R) が次のように構成される．

ステップ 1: $T :=$ **Find_Approx_Parity_ST**(F,R) とする．
ステップ 2: $T=\emptyset$ の場合，(F,R)-許容なパリティ全域木は存在しないので **return**.
ステップ 3: Tが (F,R)-許容なパリティ全域木の場合，厳密解が得られたのでこれを出力して終了．
ステップ 4: Tの奇パリティ対の一つを E_iとして，以下を実行．
(i) **Find_Parity_ST**$(F\cup\{E_i\},R)$ を再帰呼出し．
(ii) **Find_Parity_ST**$(F,R\cup\{E_i\})$ を再帰呼出し．

例 2 図 3 は例 1 と同じ問題で，子問題が生成される様子を示す．○，×は固定，禁止枝グループを示し，zは近似解の評価値である．子問題 P_4は実行不可能で，子問題 P_5で図 4 に示すパリティ全域木が得られた．

4 数値実験

節点数 101~501 の平面グラフについて，枝をランダムに対として，それぞれ 10 回ずつ解いた．表 1 はその集計で，近似解については反復回数（#Step），解の精度（Gap）と CPU 時間（秒）を，厳密解については生成子問題数と CPU 時間を示している．ただし，厳密解法は子問題数 10000 までで計算を打ち切っており，表はこの範囲で計算が終了したケースについての平均である．

近似解法は高速であるが，かなり大きいギャップが残る．厳密解法は，問題が大規模になると非常に時間がかかるケースが生じるようになる．

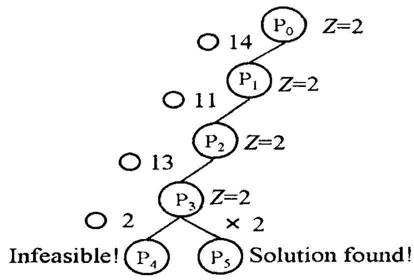

図 3: 厳密解法の動作

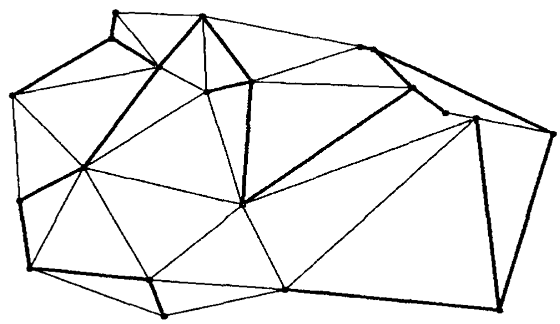

図 4: 厳密解

表 1. 実験結果

n	m	近似解			厳密解	
		#Step	Gap	CPU	#sub	CPU
101	240	25.40	7.60	0.01	31.60	0.14
201	500	52.50	14.60	0.02	187.10	1.52
301	800	80.00	23.40	0.04	215.12[a]	3.59[a]
401	1050	106.90	28.60	0.07	690.57[b]	13.32[b]
501	1370	135.70	36.00	0.11	353.71[b]	15.17[b]

[a] 計算が終了した 8 題の平均
[b] 計算が終了した 7 題の平均

5 むすび

パリティ全域木問題について，解法アルゴリズムを提示した．今後，タブー・サーチ等のより洗練された近似解法を試みてみたい．

参考文献

[1] M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman, 1979.

[2] R.K. Ahuja *et al.*, *Network Flows*. Precntice Hall, 1993.

[3] T. コルメン他.「アルゴリズム: イントロダクション」近代科学社, 1995.