3G－6

ネットワーク信頼度設計における最適／準最適2分計算木の計算複雑度

巽　久行，　古屋 貴博，　徳増 眞司
神奈川工科大学 工学部 情報工学科

1．まえがき

ネットワークアベイラビリティの2分計算木から得られるパス集合から，最小またはほぼ最小の2分計算木を生成するアルゴリズムは既に提案されている[1]。本報告では，この最適解と準最適解を与える2通りの方法に対する計算の複雑さと解の妥当性を評価する。

2．生成アルゴリズム

文献[2]の新2分計算木法より，ネットワークのパス集合が求められる。このパス集合から最適または準最適な2分計算木を再構成するアルゴリズムは，次のようになっている[1]。以下，ネットワークにおけるリンク数をNol，パス総数をNopathとし，パス集合path[i][j]を

$$\mathrm{path}[i][j]=\begin{cases}1\ (\text{パス}\#i\ \text{がリンク}\ \#j\ \text{を含むとき})\\ 0\ (\text{その他})\end{cases}$$

(但し $i=1\sim\mathrm{Nopath}$，$j=1\sim\mathrm{Nol}$）とする。パス$\#i$の長さをpf[i]，リンク$\#j$を有するパスの数をlf[j]とすると，

$$\mathrm{pf}[i]=\sum_{j=1}^{\mathrm{Nol}}\mathrm{path}[i][j],\qquad \mathrm{lf}[j]=\sum_{i=1}^{\mathrm{Nopath}}\mathrm{path}[i][j]$$

である。

最適な2分計算木を生成するアルゴリズムは，まず最初に与えられたpf[i]，lf[j]およびpath[i][j]を用いて，対象とするノード以下の各ノードの最適リンクを葉のノードから遡って順次決定することにより，そのノードにおける最適な展開リンク（これをpivotと呼ぶ）を求める（以下，これを後退探索と呼ぶ）。次に計算木の根より順次葉に向かって各ノードのpivotを決定していく（これを前進探索と呼ぶ）。最終的に2分計算木の＋系列と－系列が全て決定され，＋系列の総和よりアベイラビリティを求めて終了する。準最適化法は，上記2つの探索法にヒューリスティックスを加えて前進型にしたものである。以下に，準最適化法に用いているヒューリスティック探索法のアルゴリズムを述べる。

【ヒューリスティック探索法】

【ステップ1】 展開候補リンクの選定を行う。パス集合において，最小のpf[i]を有するパスが含むリンクで，最大のlf[j]を有するリンク$\#j$を展開候補とする。

【ステップ2】 展開候補リンクの稼働／非稼働別に，現パス集合から下位パス集合を生成する。候補リンクが稼働時状態（これをUplinkと呼ぶ）の場合，展開候補リンク$\#j$を含むパスのj列に＋記号を記入して下位パス集合を得る。候補リンクが非稼働時状態（これをDownlinkと呼ぶ）の場合，リンク$\#j$を含むパスに－記号を記入し，更にそのパスに含まれるすべてのリンクに－記号を記入して下位パス集合を得る。

【終了条件】 パス集合内の1つのパス$\#i$に含まれるリンクがすべて＋記号の場合，生成ノードは＋系列で決定され，以降のノード生成は行わない。またパス集合内のすべてのパスがすべて－記号の場合，パスはカット状態であり，生成木ノードは－系列で決定され，以降のノード生成は行わない。すべての生成ノードが＋系列か－系列で決定されたとき，生成ノードは終端(葉)となり手順は終了する。

3．最適化／準最適化アルゴリズムの計算量

3.1　最適化アルゴリズム

ネットワークNをリンク集合として，各リンクにはリンク信頼度が付与されているものとする。即ち，

$N=\{e_j\}$，$\forall e_j$；$\mathrm{Prob}(e_j)=p_j$　$(0<p_j<1)$。

以下，便宜上e_jをjと略記する。

(1)　後退アルゴリズム（bsearch）

$f(N)$を，Nの展開木の最小ノード数とし，pivot(N)を，$f(N)$に相当する，最適木の最初の展開リンク番号とする。この関係を図1に示す。ここで$f_j(N)$を，リンクjを最初に展開したときの，Nの展開木の最小ノード数とすると，

$$f(N)=\min_{j\in N}f_j(N)\tag{1}$$

$$\mathrm{pivot}(N)=\arg\left(\min_{j\in N}f_j(N)\right)\tag{2}$$

であり（但し式(2)は，$f_{j_p}(N)=f(N)$となるリンク番号j_pを示す），

$$f_j(N)=1+f(N\mid\{j\})+f(N-\{j\})\tag{3}$$

が成り立つ（図1参照）。よって，

$$f(N)=1+\min_{j\in N}\left(f(N\mid\{j\})+f(N-\{j\})\right)\tag{4}$$

となって再帰的な式となる。この式の終端条件は，

（a）　Nにパスが生成されたとき，

$f(N)=1$で，このときpivot(N)$=-1$

（b）　Nにカットが生成されたとき，

$f(N)=1$で，このときpivot(N)$=-2$

とする。この後退アルゴリズム（bsearch）は，$n=|N|$として$f(N)$を計算するための計算量を$f^{\#}(n)$とすると，式(4)より，

$$f^{\#}(n)=2n\cdot f^{\#}(n-1)\tag{5}$$

となる。ここで$f^{\#}(0)=1$であり，計算量の上界はパスやカットが展開途中で現れないと仮定する場合として，

$$f^{\#}(n)=2^n\cdot n!\cdot f^{\#}(0)=2^n\cdot n!\tag{6}$$

を得る。

A Complexity of Optimal/Quasi-optimal Binary Computation Tree for Reliability Design.
Hisayuki Tatsumi, Takahiro Furuya, Shinji Tokumasu
Kanagawa Institute of Technology

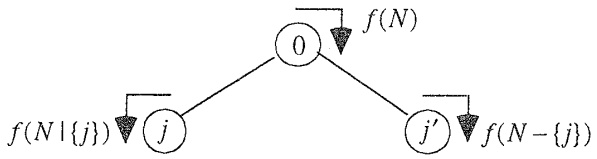

図１　展開木の最小ノード数$f(N)$の関係

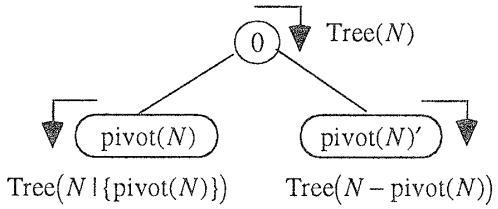

図２　最小木Tree(N)の関係

（2） 前進アルゴリズム（fsearch）

pivot(N)は前述したように，

（a） 非負のとき，Nに関して最小木を実現するための最初の展開リンク番号を与え，

（b） 負（−1または−2）のとき，木の終端ノードであることを示し，以降の展開は不要

となるのでTree(N)をNの最小木とすると，図2からこれも再帰的に生成できる。$n=|N|$として，Tree(N)を生成するための計算量を$Tree^{\#}(N)$とすると，

$$Tree^{\#}(n)=f^{\#}(n)+2\cdot Tree^{\#}(n-1) \tag{7}$$

であり，$Tree^{\#}(0)=0$であることから，

$$Tree^{\#}(n)=\sum_{k=0}^{n-1}2^k\cdot f^{\#}(n-k)+2^n\cdot Tree^{\#}(0)$$

$$=2^n\cdot\sum_{k=0}^{n-1}k! \tag{8}$$

が最適木を生成する計算量の上界として得られる。

3.2　準最適化アルゴリズム

前節3.1で述べた最適化アルゴリズムのbsearchと比較すると，ヒューリスティク探索法を用いた場合，pivot(N)の処理部分の計算量は無視できるので，$f(N)$を計算するための計算量$f^{\#}(n)$は式(5)に対して，

$$f^{\#}(n)=2\cdot f^{\#}(n-1) \tag{9}$$

となる。よって，式(6)の代わりに，

$$f^{\#}(n)=2^n\cdot f^{\#}(0)=2^n \tag{10}$$

となる。Nの準最小木Tree(N)を生成するための計算量$Tree^{\#}(N)$は，$f(N)$が求まったときにはTree(N)が作られているので，式(7)に対して，

$$Tree^{\#}(n)=f^{\#}(n) \tag{11}$$

と考えることができる。従って，式(8)の代わりに

$$Tree^{\#}(n)=2^n \tag{12}$$

が準最適化の場合の計算量の上界となる。このことより準最適化の方が，単純な計算量の比較として最適化よりも$\sum_{k=0}^{n-1}k!$倍の高速性が存在することが分かる。

4．最適2分計算木の生成例

図3に示す4ノードネットに対して，文献[2]の手法で2分計算木を生成すると，根を含めてノード数21の木が得られる。その2分計算木は，リンクの展開順を変えるとノード数を減らすことが可能であり，計算量が減少する。しかしながら，生成した木の違いによってアベイラビリティ値が変化することはない。

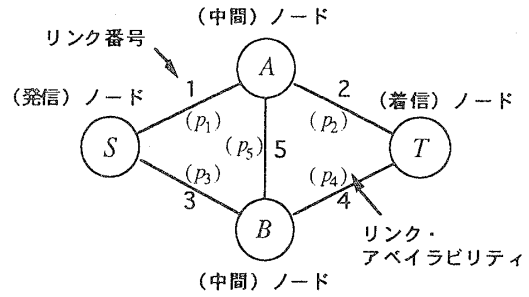

図３　４ノード・ネットワーク

図4に準最適アルゴリズムで求めた，根を含めてノード数19の2分計算木を示す。最適アルゴリズムは，展開リンクの候補が複数ある場合，後退および前進からなる全幅探索を施して最適展開リンク(pivot)を決定する。この例では，最適と準最適の結果は同じであった。また文献[1]に，文献[2]の新2分計算木法で作成した場合と，最適／準最適化法で作成した場合の，互いの計算木の生成ノード数が比較されている。

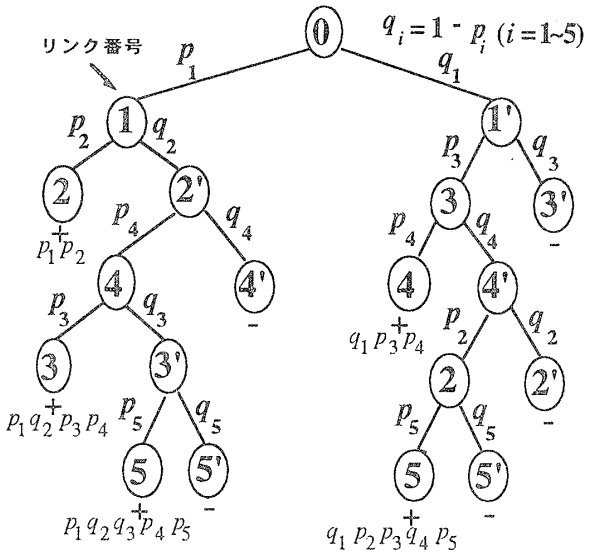

図４　図３に対する準最適(最適)な2分計算木

5．あとがき

ネットワークのパス集合から，最適ないし準最適な2分計算木を生成するアルゴリズムが提案されている。ここではこれら2通りのアルゴリズムを，計算の複雑さと解の妥当性の面で評価した結果，計算量の点から準最適化アルゴリズムの有効性を示すことができた。

参考文献

[1] 巽，古屋，徳増：信頼性設計のための最適2分計算木の生成，情報処理学会第57回全国大会講演論文集(1)，pp.1_119-1_120，1998.

[2] 古屋，巽，徳増：信頼性設計のための新2分計算木法の試作と評価，情報処理学会第57回全国大会講演論文集(1)，pp.1_117-1_118，1998.