

信頼性設計のための最適2分計算木の生成

3C-4

巽 久行, 古屋 貴博, 徳増 真司
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バス集合から、最適な2分計算木を生成するアルゴリズムを提案する。このアルゴリズムは、アベイラビリティを計算するため、最小またはほぼ最小の2分計算木を構成するので、アベイラビリティの繰り返し計算を必要とする信頼性設計にとって有効な手段を提供する。

2. 生成アルゴリズム

文献[1]の新2分計算木法より、ネットワークのバス集合が求められる。本報告は、このバス集合から準最適または最適な2分計算木を再構成することを目的とする。

ネットワークにおけるリンクの数を N_{link} 、バス総数を N_{path} とし、バステーブル $\text{path}[i][j]$ を以下のように定義する（但し、 $i = 1 \sim N_{\text{path}}$, $j = 1 \sim N_{\text{link}}$ とする）。

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

また、バス $\#i$ の長さを $\text{pf}[i]$ 、リンク $\#j$ を有するバスの数を $\text{lf}[j]$ とすると、

$$\text{pf}[i] = \sum_{j=1}^{N_{\text{link}}} \text{path}[i][j], \quad \text{lf}[j] = \sum_{i=1}^{N_{\text{path}}} \text{path}[i][j]$$

である。本報告ではバステーブルを図1のように記す。

link#	リンク				
path#	1	…	j	…	N_{\text{link}}
バス	path[i][j]				
Nopath					pf[i]

図1. バステーブル

最適な2分計算木を生成するアルゴリズムは動的計画法(DP)を用いており、まず最初に与えられた $\text{pf}[i]$, $\text{lf}[j]$ および $\text{path}[i][j]$ 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を述べる。

Generation of an Optimal Binary Computation Tree for Reliability Design.

Hisayuki Tatsumi, Takahiro Furuya, Shinji Tokumasu
Kanagawa Institute of Technology

【ヒューリスティック探索法】

【ステップ1】最初に展開候補リンクの選定を行う。図1に示すバステーブルにおいて、最小の $\text{pf}[i]$ を有するバスが含むリンクで、最大の $\text{lf}[j]$ を有するリンク $\#j$ を展開候補とする（但し、候補が複数存在した場合、リンクの名前順で展開する）。

【ステップ2】次に展開候補リンクの稼働／非稼働別に、現バステーブルから下位バステーブルを生成する。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 $\#j$ を含むバスの j 列に+記号を記入して下位バステーブルを得る。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 $\#j$ を含むバス（ pf 列）に-記号を記入し、更にそのバスに含まれるすべてのリンクに-記号を記入して下位バステーブルを得る。

【終了条件】バステーブル内の1つのバス $\#i$ 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 ($\text{pf}(i) = '+'$)。以降のノード生成は行わない。またバステーブル内のすべてのバスがすべて-記号の場合、バスはカット状態であり、生成木ノードは一系で決定され、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2分計算木の生成例

図2に示す4ノードネットに対して、文献[1]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。

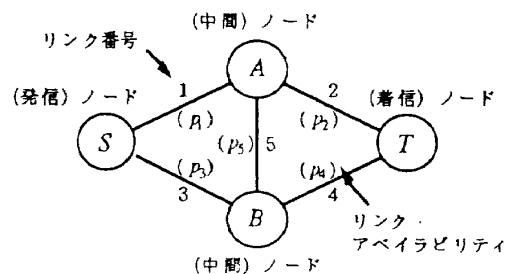


図2. 4ノード・ネットワーク

本アルゴリズムを用いて、図2の2分計算木の準最適解を求めてみる。以下、木の根からのノードの深さを $depth$ と記し、 $depth$ を増やしながらすべての生成ノードが+系列か-系列に決定されるまで繰り返す。

【 $depth = 0$ (即ち根)の場合】図2に文献[1]の手法を適用すると、リンク #1 ～ #5 を要素としたリンク集合からなる4つのバス（{1, 2}, {1, 4, 5}, {3, 4}, {2, 3, 5}）が得られる。この集合で構成される初期バステーブルを、図4.0に示す。ここで最小の $\text{pf}[i]$ （この場合は2）

	1	2	3	4	5	pf[i]
1	○	○				2
2	○		○	○		3
3		○	○			2
4	○	○		○		3
lf[j]	2	2	2	2	2	

図 4.0. Case: 0

を有するバスは #1 と #3 であり、このバスが含むリンクで最大の lf[j] (この場合は 2) を有するリンクは #1, #2, #3 および #4 であり、リンクの名前順からリンク #1 を選択して木を展開する。

【depth=1 の場合】図 4.0 からリンク #1 を選択して生成した Uplink 側バステーブルを図 4.1(a) (図はリンク #1 を含むバスに + 記号を記入して得られる) に、Downlink 側バステーブルを図 4.1(b) (図は、リンク #1 を含むバスに - 記号を記入し、更にそのバスに含まれるすべてのリンクに - 記号を記入して得られる) に示す。選択されたリンクは、これ以降、Uplink 側および Downlink 側の双方のバステーブルから除外される。

	1	2	3	4	5	pf[i]
1	+	○				1
2	+		○	○		2
3		○	○			2
4	○	○		○		3
lf[j]	0	2	2	2	2	

図 4.1(a). Case: 1

	1	2	3	4	5	pf[i]
1	-	-				-
2	-		-	-		-
3		○	○			2
4	○	○		○		3
lf[j]	0	1	2	1	1	

図 4.1(b). Case: 1'

図 4.1(a)の場合、最小の pf[i] (この場合は 1) を有するバスは #1 であり、このバスが含むリンクで最大の lf[j] (この場合は 2) を有するリンクは #2 であり、次の depth=2 ではリンク #2 を選択して木を展開する。図 4.1(b)の場合、最小の pf[i] (この場合は 2) を有するバスは #3 であり、このバスが含むリンクで最大の lf[j] (この場合は 2) を有するリンクは #3 であり、次の depth=2 ではリンク #3 を選択して木を展開する。

【depth=2 の場合】図 4.1(a)からリンク #2 を選択して生成した木の Uplink 側バステーブルを図 4.2(a)に、Downlink 側バステーブルを図 4.2(b)に示す。また、図 4.1(b)からリンク #3 を選択した Uplink 側テーブルを図 4.2(c)に、Downlink 側テーブルを図 4.2(d)に示す。

	1	2	3	4	5	pf[i]
1	++					+
2	+		○	○		2
3		○	○			2
4	+	○		○		2
lf[j]	0	0	2	2	2	

図 4.2(a). Case: 1 2

	1	2	3	4	5	pf[i]
1	--					-
2	+		○	○		2
3		○	○			2
4	-	-	-	-		-
lf[j]	0	0	1	2	1	

図 4.2(b). Case: 1 2'

	1	2	3	4	5	pf[i]
1	- -					-
2	-		- -	- -		-
3		+	○			1
4	○	+		○		2
lf[j]	0	1	0	1	1	

図 4.2(c). Case: 1 3'

	1	2	3	4	5	pf[i]
1	--					-
2	-		- -	- -		-
3		-	- -	-		-
4	-	-	-	-		-
lf[j]	0	0	0	0	0	

図 4.2(d). Case: 1 3'

図 4.2(a)の場合、生成ノード 2 は + 系列で決定され、このバステーブルからの以降の木生成は終了した。図 4.2(d)の場合、バステーブルはすべて “-” であることから、すべてのバスはカット状態であり、生成ノード 3 は - 系列で決定され、このバステーブルからの以降の木生成は終了した。図 4.2(b)および図 4.2(c)については、更にリンクを選択して depth=3 の木を生成する。これを繰り返して、+ 系列か - 系列で一意にすべての葉が決定されたとき、アルゴリズムは終了する。

図 5 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、後退及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。

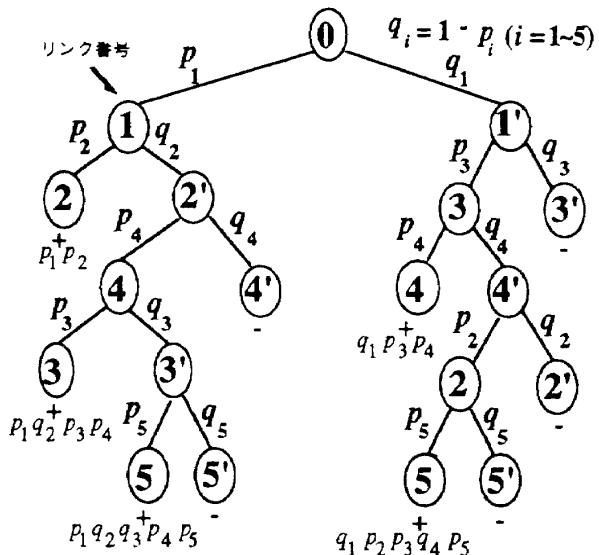


図 5. 図 3 に対する準最適な 2 分計算木

4. 実行結果の比較

最適化前と本アルゴリズムの比較を以下に示す。

表 1. 2 分計算木の生成ノード数比較

アルゴリズム ネット サイズ	2 分計算木 有向 無向	準最適化法 有向 無向	最適化法 有向 無向
N(4,5)	17 21	15 19	15 19
N(6,8)	49 69	39 57	39 57
N(8,13)	133 171	131 157	-
N(9,12)	145 345	91 239	-
N(11,21)	4339	755	-
N(16,30)	83169	9039	-

5. あとがき

ネットワークのバス集合から、最適な 2 分計算木を生成するアルゴリズムを提案した。今後の課題として、アルゴリズムの計算量を評価すること等が挙げられる。

参考文献

- [1] 古屋, 義, 德増: 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, 3C-3, 1998.10(発表予定).