

3 C - 4

神奈川工科大学 工学部 情報工学科

2. 生成アルゴリズム

図1. パステープル

図2. 4ノード・ネットワーク

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

	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

を有するパスは #1 と #3 であり、このパ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 を含むパスに + 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) を有するパスは #1 であり、このパ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) を有するパスは #3 であり、このパ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、すべてのパスはカット状態であり、生成ノード 3' は一列で決定され、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。

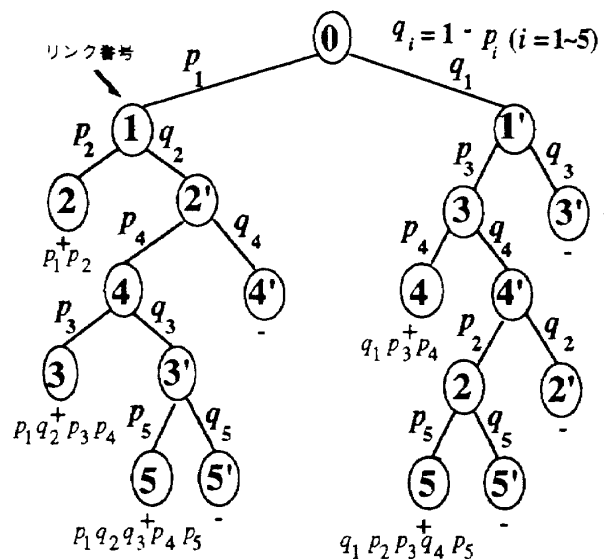


図 5. 図 3 に対する準最適な 2 分計算木

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	2 分計算木	準最適化法	最適化法
サイズ	有向 無向	有向 無向	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

ネットワークのパス集合から、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, 巽, 徳増: 信頼性設計のための新 2 分計算木の試作と評価, 情報処理学会第 57 回全国大会, 3C-3, 1998.10(発表予定).