

信頼性設計のための新2分計算木法の試作と評価

3C-3

古屋 貴博, 巽 久行, 徳増 眞司
 神奈川工科大学

1. まえがき

情報システムが益々高度化するに伴い、インターネットやイントラネットに対するエンドユーザレベルの依存度が急激に増大している。このため今までのようなサービス形態から、いかにして早く接続できるかというサービス形態が重要になりつつある。そこで、本研究は現在のサービス形態をより一層良質なものにすべく新アルゴリズムを提案する。アベイラビリティ計算効率化のための新しい状態空間分解法のアルゴリズムを用い、プログラム化して実験、検討を行った結果、本アルゴリズムが十分な信頼性の基で早くネットワーク接続を行うために有効なツールとなり得る事実を、従来の方法の中で最も効率的である Dotson に関する論文^[1]に基づいて追試を行った結果を比較・検討に用い、実証した。

2. 状態空間分解法のアルゴリズム

ネットワークの信頼性を見つける一つの方法として、状態空間分解法がある。これは、各要素が動作か故障かであるかを考慮することによって行われる。いま、ネットワークNのアベイラビリティをA(N)とすると、これは次の式で表される。

$$A(N) = P(C) \cdot A(N|C) + (1-P(C)) \cdot A(N-C)$$

- ここで、P(C) : Cが稼働している確率
- N : 対象とする一つのネットワーク
- C : ネットワークNの構成要素である一つのネットワーク
- A(N|C) : NのうちCが稼働している場合のアベイラビリティ
- A(N-C) : NのうちCが故障している場合のアベイラビリティ

この式は、ネットワークを排他的な二つの部分に分けることによって、ネットワーク・アベイラビリティを計算するものであり、これを状態空間分解法という。

3. 新アルゴリズム

本論文で提案するアルゴリズムはステップ1からステップ4で構成されている。以下、図1を用いて説明する。

Prototyping of a New Binary Computation Tree for Reliability Design.

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

(1) ステップ1

ネットワークをグラフG(N, L, P)として入力する。但し、Nはノード集合、Lをリンク集合、Pを各辺に付加されているアベイラビリティ集合とする。

(2) ステップ2

前進ラベリングと後退ラベリングを各ノードに持たせ

る。前進ラベリングではT以外のノードkに対するSからの距離 i_k をラベリングし、全てのノードにラベリングをし終えたらTのラベルに無限大を挿入する。これに対し、後退ラベリングでは、S以外のノードkに対するTからの距離 j_k としラベリングを終えたらSに無限大を挿入する。ステップ2の役割は、リンク展開の優先順位を決定することである。

(3) ステップ3

ステップ2において生成したラベルによりリンクの順序を決定しリンクテーブルに登録する。

[ノードnのリンクテーブル生成ルール]

(1) 登録

$n_i \neq S$ ならばリンク l_i を登録する

$n_i = S$ ならばリンク l_i を登録しない

(2) 順位 (Sよりリンク先のノードのjで比べる)

l_1, l_2 が登録される物とするとき

$i_1 > i_2$ なら $l_1 < l_2$

$i_1 = i_2$ で $j_1 < j_2$ なら $l_1 < l_2$ (l_1 が l_2 より高順位)

(4) ステップ4

アベイラビリティ計算木を生成する。リンクテーブルを用いて計算木を生成していくと同時に、各分岐点には、展開候補テーブルを生成していく。(各分岐点の展開候補テーブルを逐次、更新、再生していくことで展開していく。)

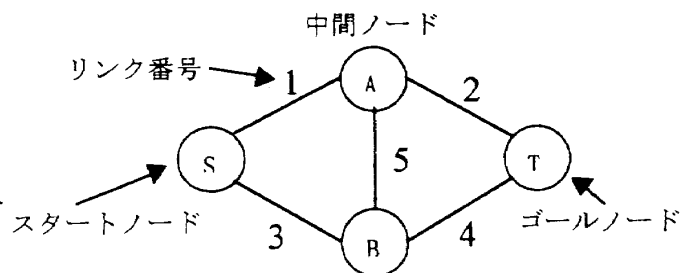


図1 4ノードネットワーク

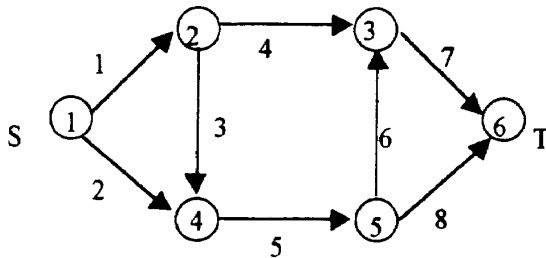
4. 実験結果

新アルゴリズムを用いてプログラムを作成した際に、改良型 Dotson 法の追試を行い、比較・検討が可能になるように、同じ実験データを用いた。次にプロセッサ時間の計測、全パス数の計測ができるように Borland C++ によってプログラム化し、Dos/v パソコン上で数値実験を行った。(Gateway2000, 200メガヘルツ)

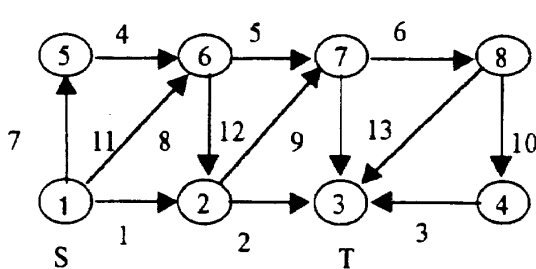
<実験>

ノード数 4,6,8,9,11 のネットワークについて計算木の生成を行い、ネットワークの信頼度を計算した。但し、各リンクの信頼度は(4) 11ノードネットワークを除いて、すべて0.9である。

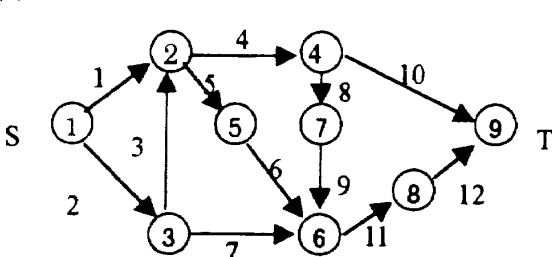
(1) 6ノードネットワーク



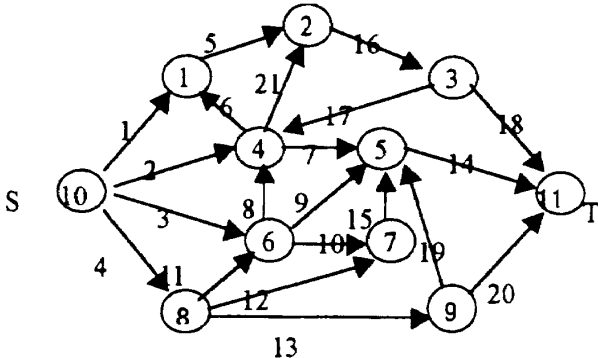
(2) 8ノードネットワーク



(3) 9ノードネットワーク



(4) 11ノードネットワーク



($r_6=0.81, r_3=r_{11}=r_{12}=r_{18}=0.981$ その他のリンクは $r=0.9$ とした。)

<改良型 Dotson との比較>

改良 Dotson の提案したアルゴリズムと本アルゴリズムとの比較を表 1, a, b, に示す。

表 1 a. 無向ネットワーク数値比較表

	リンク数	パスの数	Reliability	no of Multi	no of Adj	処理時間	
4 node	新アルゴ	5	4	0.97848	20	11	0.16
	Dotson	5	4	0.97848	43	11	0.05
6 node	新アルゴ	8	8	0.96698	68	35	0.22
	Dotson	8	8	0.96698	260	41	0.05
8 node	新アルゴ	13	23	0.99495	170	86	0.39
	Dotson	13	23	0.99495	669	79	0.06
9 node	新アルゴ	12	13	0.96486	344	173	0.38
	Dotson	12	13	0.96486	1380	154	0.11
11 node	新アルゴ	21	151	0.99887	40952	20637	2.03
	Dotson	21	151	0.99887	145699	8484	5.61

表 1 b. 有向ネットワーク数値比較表

	リンク数	パスの数	Reliability	no of multi	no of Adj	処理時間	
4 node	新アルゴ	5	3	0.97119	16	9	0
	Dotson	5	3	0.97119	32	9	0
6 node	新アルゴ	8	5	0.95208	48	25	0.11
	Dotson	8	5	0.95208	138	25	0
8 node	新アルゴ	13	18	0.99473	132	67	0.28
	Dotson	13	18	0.99473	572	70	0.06
9 node	新アルゴ	12	7	0.94892	144	73	0.22
	Dotson	12	7	0.94892	576	74	0.05
11 node	新アルゴ	21	18	0.99719	4338	2170	0.44
	Dotson	21	18	0.99719	15300	1147	0.49

表 1 a. 1 b. の no of multi の欄を比較してみると、Dotson の提案したアルゴリズムよりも我々の提案したアルゴリズムのほうが数値が小さくなっている。特に 11ノードのネットワークの数値は改良型 Dotson の方が、倍以上に増えている、これは乗算の回数を比較しており、数値が小さいほうが処理時間の短縮ができるので我々の提案したアルゴリズムのほうが早いと確認できた。新アルゴリズムは接点数の増加率を低く抑えられるので、さらに大きなネットワークに効果が発揮できると考える。

6. あとがき

ネットワークアベイラビリティの計算に関し前進・後退ラベリングに基づくリンクテーブルの生成により、2分木展開の効率化が可能となり、新アルゴリズムの有効性が確認できた。今後の課題としては、処理時間のより正確な計測を行うことと、またネットワークサイズに対する負荷性能変化を明らかにするため、更に大きなネットワークに対する実験を行うことなどがある。

7. 参考文献

【1】 Y.B Yoo, Narsi Deo, A comparison of Algorithms for Terminal Pair Reliability