

2 端子ネットワーク信頼性解析と近似解析法

荒木智行・山本富士男

情報工学科

Analyzing Two-terminal Networks Availability and their Approximate Estimation

Tomoyuki Araki , Fujio Yamamoto

Abstract

This paper describes a new algorithm to calculate availabilities of communication networks by using principle of optimality of dynamic programming (DP). It is well known that state space decomposition method to calculate availabilities of communication networks is one of problems of NP-hard. Therefore, it is important to reduce computational complexity. The proposed algorithm gives a procedure to save the size of binary tree on the way of state space decomposition. In the algorithm, the calculation of a part consisting of high network availability is taken place prior to other parts.

From the feature of this algorithm, this paper also proposes the method to calculate the lower bound of network availability, and decomposition and reduction method to apply this algorithm to large scale networks.

Key Words: Network availability, Network of mesh type, Dinamic programing, State space decomposition

1 まえがき

通信ネットワークに対して社会・経済活動の依存度が増すにつれて、ネットワークの経済性はもとより、信頼性に対する関心が高まっている。

ところで、いかなる設備も突発的な故障の発生を完全に避けることはできない。このため、一般に冗長設計がとられている。ネットワークの場合、ノードとリンクそれぞれに対して冗長設計がとられている。この場合、各部の冗長設計によってネットワーク全体の信頼性が、どのように改善されるのか、投資効率という面からも、その定量的評価が重要になる。

通常、ネットワークにおける2地点間信頼性 (two-terminal reliability) の問題は、ネットワークの問題をグラフの問題としてとらえ解析がなされている。その代表的な手法として状態空間分解法 (state space decomposition) がある。一般的にこの手法では状態空間分解を行う際、2分木を作成するが、分解を行うリンクの順番の指定の仕方により、木のサイズが大きくなったり小さくなったりする。この問題は、木の点の数に依存して

指数関数的に計算量が増大するNP困難な問題であることが良く知られており、与えられたネットワークに対して、点の数の少ない木を作成することが、大規模なネットワークの信頼性評価では重要である。

著者らは、状態空間分解における木のサイズ (点の数) を有効に削減できる実用的アルゴリズムを提案している [8]。提案アルゴリズムの特徴としては、単にサイズの小さい2分木を作成して信頼性計算を行うことのみを目的とするのではなく、より高いネットワーク・アベイラビリティを構成する部分を優先して計算している点である。それを実現するために、2分木作成の際の枝出しにおいて、ネットワークのアベイラビリティをより大きくするよう、リンクの選択を行う。これはダイナミック・プログラミング (dynamic programming : DP) の最適性の原理により実現する。現実のネットワークでは、日々、形状・仕様の変更がなされており、よりアベイラビリティの高いネットワークを構成するようにすることは重要である。

提案アルゴリズムの特徴を踏まえて、本論文では更に、ネットワーク・アベイラビリティの下界の計算法につい

でも提案する。また、計算困難さを更に軽減するために、提案アルゴリズムを大規模ネットワークに適用する際のネットワークの分解法および縮退法についても提案する。

以上のことから提案アルゴリズムがメッシュ型のネットワークの信頼性の定量的評価に有効であることを示す。

2 通信ネットワークの信頼性の評価尺度

通信ネットワークの信頼性を定量的に評価するに当たって、信頼性の尺度を明らかにする必要がある。

通信ネットワークに関する重要な品質基準として接続品質、伝送品質、信頼性品質がある [1]。接続品質や伝送品質は「機器が正常なときのサービス特性」を扱い、信頼性品質は、「各種原因（ハードウェア故障、ソフトウェアバグ、誤操作等）によって発生する機器の故障によるサービス特性の劣化・回復特性」を扱うものである。信頼性品質は、利用の満足度、社会の要請、経済性などを総合的に考慮して定められる。この品質水準を維持するために各通信設備に対して所定の信頼性規格値が配分され、通信網の設計、建設、保守が行われている。

通信ネットワークの信頼性品質の評価尺度として我が国では不稼働率がとられている。ここで、障害に対する不稼働率とは、呼が各設備の障害に出会い、正常に処理できない度合いを示す。この不稼働率の補数が信頼性尺度のアベイラビリティである。アベイラビリティは、修理系が規定の時点で機能を維持している確率、またはある期間中に機能を維持する時間の割合と定義されている [2]。

単一系のアベイラビリティ A は、次式で定義される。

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \quad (1)$$

ここで、MTBF は平均故障間隔を表し、MTTR は平均修理間隔を表す。

よって、アベイラビリティと不稼働率の関係は、次のようになる。

$$A = 1 - \text{不稼働率} \quad (2)$$

3 アベイラビリティの計算法

これまで、多数の構成要素からなるシステムのアベイラビリティの計算法は、システムを直列系と並列系に分解して、それぞれに次に示す計算式で計算を行い、全体を合成する方法が一般的といえる [3, 4]。

(1) 直列系

直列系システムのアベイラビリティを A とし、各構成要素のアベイラビリティを A_i ($i = 1, 2, \dots, n$) とすると、直列系のアベイラビリティは、次式で与えられる。

$$A = \prod_{i=1}^n A_i \quad (3)$$

(2) 並列系

n 個の並列冗長システムのアベイラビリティ A は、次式で与えられる。

$$A = 1 - \prod_{i=1}^n (1 - A_i) \quad (4)$$

しかしながら、この方法でメッシュ型ネットワークのアベイラビリティを計算することは、直列・並列の定義そのものが困難なため、必ずしも容易ではない。このため以下では、メッシュ型ネットワークに対する実用的なアベイラビリティの計算法について述べる。

なお、以下の検討では、ノード障害は無視できるものとして、リンク障害のみを対象にネットワーク・アベイラビリティを計算する。

ここで、計算法として、信頼性多項式 ((5) 式) の性質に基づく状態空間分割法によることとする。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 [5]。

$$A(N) = P(C) \cdot A(N|C) + (1 - P(C)) \cdot A(N - C) \quad (5)$$

ここで、 $P(C)$: C が稼働している確率、 N : 対象としてい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リンク、 $(N|C)$: N のうち C が稼働している場合、 $(n - C)$: N のうち C が故障している場合、とそれぞれする。

(5) 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。

例 1 図 1 の 4 ノードのネットワークについて、状態空間分解法でアベイラビリティを求める。通信は発信ノード S から、着信ノード T に対して行われる。 A, B は、中継ノードを示す。各リンクにはリンク番号 (1, 2, 3, 4, 5) が付してある。ここで、ノード故障はないものとする。各リンクのアベイラビリティを、ここでは簡単化のため、いずれも 0.9 とする。この場合の状態空間分解 (A) を図 2 に示す。また、これに基づくアベイラビリティの計算例

表 1: 状態空間分解法による計算結果

状態	リンク表示		確率	最終状態	アベイラビリティ	不稼働率
	アップ側	ダウン側				
0	00000	00000	1.0		.00000	.00000
1	00001	00000	.9		.00000	.00000
12	00011	00000	.81	+	.81000	.00000
1'	00000	00001	.1		.81000	.00000
12'	00001	00010	.09		.81000	.00000
1'2	00010	00001	.09		.81000	.00000
12'3	00101	00010	.081		.81000	.00000
1'23	00110	00001	.081		.81000	.00000
12'34	01101	00010	.0729	+	.88290	.00000
1'234	00111	00001	.0729	+	.95580	.00000
1'2'	00000	00011	.01		.95580	.00000
12'3'	00001	00110	.009		.95580	.00000
1'23'	00010	00101	.009	-	.95580	.00900
1'2'3	00100	00011	.009		.95580	.00900
12'34'	01101	00010	.0081	-	.95580	.01710
12'3'4	01001	00110	.0081		.95580	.01710
1'234'	01001	00110	.0081		.95580	.01710
1'2'34	01100	00011	.0081	+	.96390	.01710
12'3'45	11001	00110	.00729	+	.97119	.01710
1'234'5	10110	01010	.00729	+	.97848	.01710
1'2'3'	00000	00111	.001	-	.97848	.01810
12'3'4'	00001	01110	.0009	-	.97848	.01900
1'2'34'	00100	01011	.0009	-	.97848	.01990
12'3'45'	01001	01101	.00081	-	.97848	.02071
1'234'5'	00110	11001	.00081	-	.97848	.02152

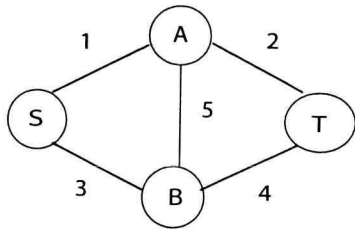


図 1: 4 ノードネットワーク

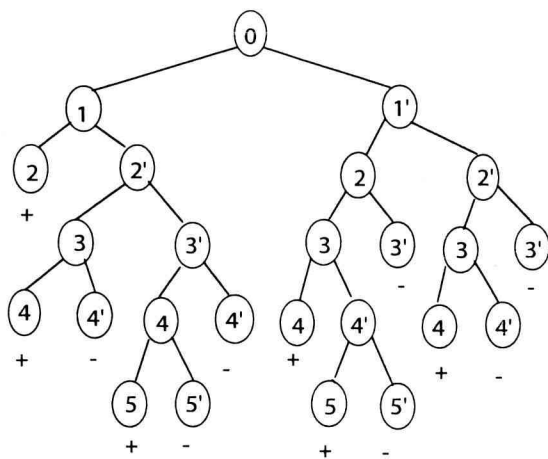


図 2: 状態空間分解 (A)

を表 1 に示す。これからアベイラビリティは, 0.97848, 不稼働率は, 0.02152 となる。この場合の分解の点の数は, 24 個となっていることがわかる。これに対して, 少し分解法を変えると, 図 3 の状態分解 (B) のように点の数は 18 となり, アベイラビリティは同じ値を与えている [5]。しかしながら, このように有効な分解の順番を決めるための手段は, 筆者らの知る限り, 現在のところ明らかに示めされていない [5]。

4 アベイラビリティの計算アルゴリズム

状態空間分解での点の数を効率的に少なくするためのメッシュ型ネットワーク・アベイラビリティの計算アルゴリズムを本節で提案する。

状態空間分解法によって 2 分木を作成し, アベイラビリティを計算する手法はメッシュ型のネットワークに対して有効な手段であるが, 計算量の立場から見ると, 2 分木の点の数に対して計算量が指数関数的に増加する NP 困難な問題である。そこで一つの木に含まれる点の数を削減することが重要な問題となる。2 分木の点の数を削減するには以下の二つの方法が考えられる。

- (1) 分解するリンクの順番を 2 分木のサイズが大きくな

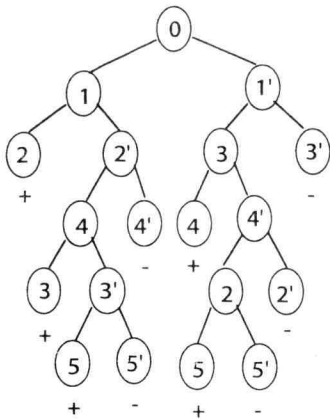


図 3: 状態空間分解 (B)

らないよう効率的に選択する.

- (2) 対象とするネットワークを、複数のネットワークに分解し、それぞれのアベイラビリティを求め、それぞれを一つのリンクと見なし (縮退), 次いで、それらを用いてネットワークを合成し、もとのネットワークのアベイラビリティを求める.

(2) については、6. で述べる. 本節で提案するアルゴリズムは、このうち (1) を DP の最適性の原理を用いて、2 分木の構造の冗長性を減らし、かつ、ネットワーク・アベイラビリティの高い構成部分を優先して取り上げるものである.

木の冗長性を減らすためのアルゴリズム上の更なる工夫としては、以下の二つを折り込む.

(Cond 1) なるべく木の根に近いレベルで S から T への経路 (path) を確立させる.

(Cond 2) なるべく木の根に近いレベルでネットワークの切断 (cut) を確立させる.

計算の原理

- (1) 各ノードにノードテーブルを用意する. ノードテーブルには、ネットワーク・トポロジーに基づいて各ノードにつながっている次リンクに関して接続リンク数、リンク番号、リンクの重み (アベイラビリティ) の情報を持つものとする.
- (2) ネットワークの発信ノード S から着信ノード T までの経路上のリンクの選択は、DP の最適性の原理

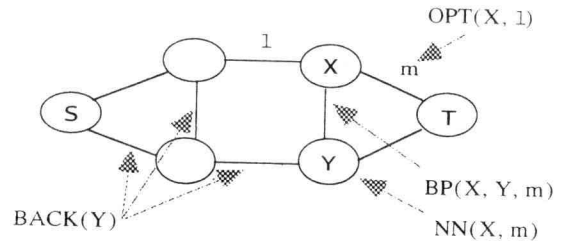


図 4: 記号の説明

により、最大のアベイラビリティが達成できる経路 (最適経路) をあらかじめ選ぶ. このためネットワーク上の段階を定義する必要がある [6]. これを定式化すると以下ようになる. n 段階でのアベイラビリティ最大経路 (最適アベイラビリティ) を $A_n(a)$ とすると、

$$A_n(a) = \max_{a_1} a_1 \cdot A_{n-1}(a|a_1) \quad (6)$$

ここで a は状態を表し、 $a = a_1 a_2 \dots a_n$ であり、 a_n は段階 n で選択されるリンクのアベイラビリティである.

- (3) 複数候補の優先条件: 選択するリンクの候補が複数ある場合は以下の優先順位に従って選択する.
- (i) 目標ノード (前進計算の場合は着信ノード T , 後退計算の場合は発信ノード S) により近い段階に接続するもの. (前述の (Cond 1), (Cond 2) の条件)
 - (ii) 以上で決められない場合はランダムに選択する.
- (4) 木の終端条件:

- S から T へ至る経路 (path) ができる場合は + 終端とする.
- S から T へ至る経路がすべて切断 (cut) される場合は - 終端とする.
- + 終端でも - 終端でもない場合はオープンである.

アルゴリズム
記号の説明 (図 4)

$OPT(X, \ell)$ リンク ℓ の終点ノード X を始点とする最適経路上のリンクの集合. ただし, $X = S$ のときは $\ell = 0$ とする.

$NN(X, m)$ リンク m の始点ノード X と, リンク m 以外のオープンなリンクで直接接続されているノードの集合.

$BP(X, Y, m)$ $Y \in NN(X, m)$ のとき, ノード X とノード Y を接続しているオープンなリンクの集合.

$BACK(Y)$ ノード Y から見て, 発信ノード S 側にあるオープンなリンクの集合.

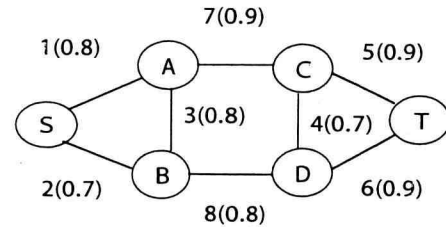


図 5: 6 ノードネットワーク

となり, 同様の操作を一終端した点に行ったものが不稼働率となる.

(Step 1) 発信ノード S において $OPT(S, 0)$ から一つのリンクを選び, 木の根より枝出しをし, アップ側の点とダウン側の点を作る.

(Step 2) 木の枝のアップ側の点では, その点に対応するリンク ℓ_i の終点ノードが X であるとき, $OPT(X, \ell_i)$ から次なる枝出しの候補を選び出し, 枝出しを行っていく. アップ側の点では順次, 同様の方法で着信ノード T に向かって前進して枝出しを行い, 木の終端条件を満たすまで続ける.

(Step 3) ノード X を始点に持つリンク m_j に対応する木の枝のダウン側の点では $NN(X, m_j)$ を求める. もし, $NN(X, m_j) = \phi$ であれば一終端である. また, $NN(X, m_j) \neq \phi$ のときは $NN(X, m_j)$ から一つのノード Y を選び出す. このとき $BP(X, Y, m_j)$, $BACK(Y)$ も同時に求める. そしてノード Y より $OPT(Y, \ell_k)$ (ただし, $\ell_k \in BP(X, Y, m_j)$) のリンクから枝出しを行い, 着信ノード T に接続するリンクまで Step 2 と同様な操作で枝出しを行った後, 最後のリンクのアップ側の点から $BP(X, Y, m_j)$ のリンクに対しても + 終端するリンクになるまで枝出しを行っていく. $BACK(Y) \neq \phi$ の場合, さらに + 終端したリンクのダウン側の点の状態では, まだオープンなリンクが残っている. この点からは, $BACK(Y)$ のリンクに対し後退して枝出しを行う.

(Step 4) Step 2, 3 を木全体の終端条件が決定するまで繰り返す.

(Step 5) + 終端した点について, 終端から根に至るまでのすべての確率を掛け合わせたものを求め, それらを加えたものがネットワーク・アベイラビリティ

(アルゴリズム終)

例 2 図 5 に示す 6 ノードの場合で, リンクのアベイラビリティがそれぞれ異なる場合を提案アルゴリズムで解くことにする. 図の上で, リンク上の 1 (0.8) はリンク番号 1, アベイラビリティが 0.8 を示す. もし, 状態空間分解を単純にリンク番号順に行ったとすると, この場合, 点の数は 229 となる. 提案アルゴリズムによる計算結果は, 図 6, 図 7, 表 2 のようになり, 点の数 58, ネットワーク・アベイラビリティは 0.89536, 不稼働率は 0.10464 となる.

5 ネットワーク・アベイラビリティの下界

提案アルゴリズムにおいて, Step 3 を取り除いた簡略化されたアルゴリズムを考えると, 最適性の原理によって選ばれた最適経路のみからなるネットワークのアベイラビリティが求められる. ここで選ばれた経路は, 各々アベイラビリティの高いリンクからなり, ネットワーク・アベイラビリティの下界 (lower bound) を形成していると考えられる. この場合, 状態空間分解によって作成される 2 分木の点の数は大幅に減少するので, ネットワーク・アベイラビリティの近似値を, 下界として求め評価する際に有効な手段となる.

6 大規模ネットワークの分解法

状態空間分解法によるネットワーク・アベイラビリティの計算は NP 困難な問題であるため, 4. で既に述べたよ

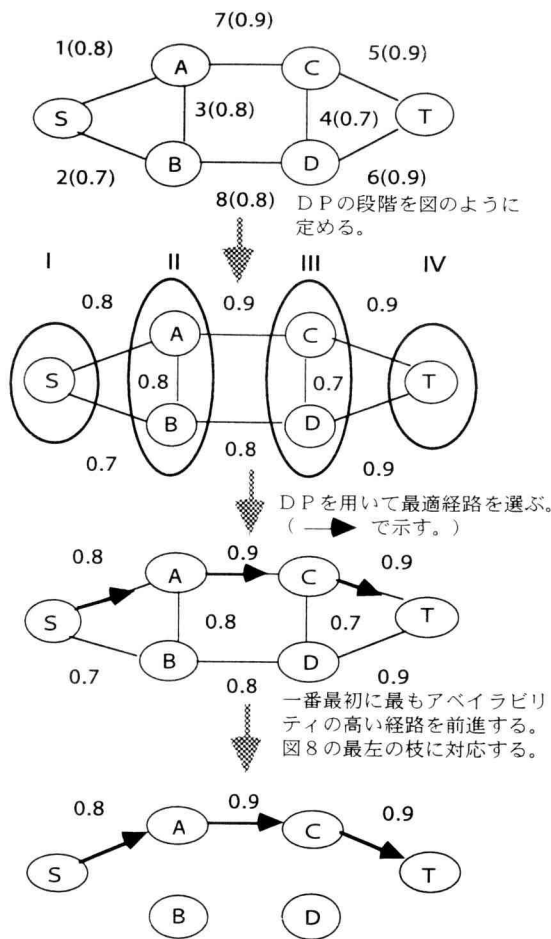


図 6: DPに基づく最適経路選定

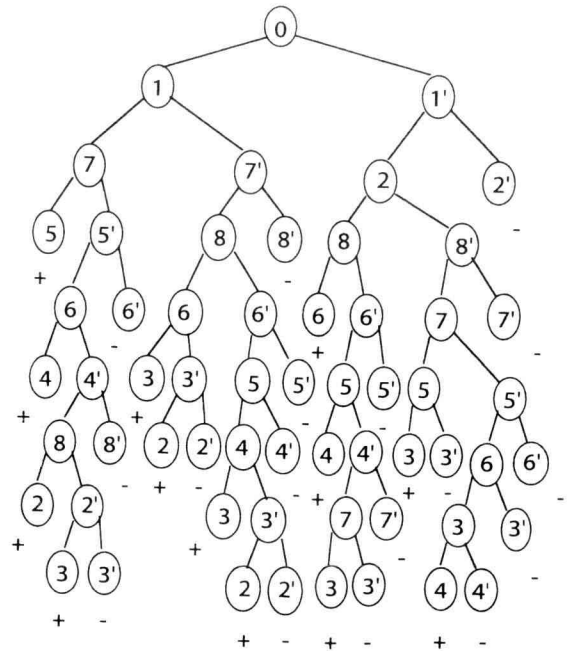


図 7: 6ノードネットワークの状態空間分解

表 2: 状態空間分解法に基づく計算結果

状態	リンク表示 アップ側 ダウン側	確率	最終 状態	アベイラ ビリティ	不稼働率
175	10001010 00000000	.64800	+	.64800	.00000
175' 64	10000110 00001000	.04536	+	.69336	.00000
175' 64' 82	11000111 00011000	.01089	+	.70425	.00000
175' 64' 82' 3	10100111 01011000	.00373	+	.70798	.00000
175' 64' 82' 3'	10000111 01111000	.00093	-	.70798	.00093
175' 64' 8'	10000110 00011001	.00389	-	.70798	.00482
175' 6'	10000010 00001100	.00720	-	.70798	.01202
17' 863	10100101 00000010	.04608	+	.75406	.01202
17' 863' 2	11000101 00100010	.00806	+	.76212	.01202
17' 863' 2'	10000101 01100010	.00346	-	.76212	.01548
17' 86' 543	10111001 00000110	.00323	+	.76535	.01548
17' 86' 543' 2	11011001 00100110	.00056	+	.76591	.01548
17' 86' 543' 2'	10011001 01100110	.00024	-	.76591	.01572
17' 86' 54'	10001001 00010110	.00173	-	.76591	.01745
17' 86' 5'	10000001 00001110	.00064	-	.76591	.01809
17' 8'	10000000 00000011	.01600	-	.76591	.03409
1' 286	01000101 10000000	.10080	+	.86671	.03409
1' 286' 54	01011001 10000100	.00706	+	.87377	.03409
1' 286' 54' 73	01101011 10010100	.00218	+	.87595	.03409
1' 286' 54' 73'	01001111 10110000	.00054	-	.87595	.03463
1' 286' 54' 7'	01001001 10010110	.00030	-	.87595	.03493
1' 286' 5'	01000001 10001100	.00112	-	.87595	.03605
1' 28' 753	01101010 10000001	.01814	+	.89409	.03605
1' 28' 753'	01000010 10100001	.00454	-	.89409	.04059
1' 28' 75' 634	01110110 10001001	.00127	+	.89536	.04059
1' 28' 75' 634'	01100110 10011001	.00054	-	.89536	.04113
1' 28' 75' 63'	01000110 10101001	.00045	-	.89536	.04159
1' 28' 75' 6'	01000010 10001101	.00025	-	.89536	.04189
1' 28' 7'	01000000 10000011	.00280	-	.89536	.04464
1' 2'	00000000 11000000	.06000	-	.89536	.10464

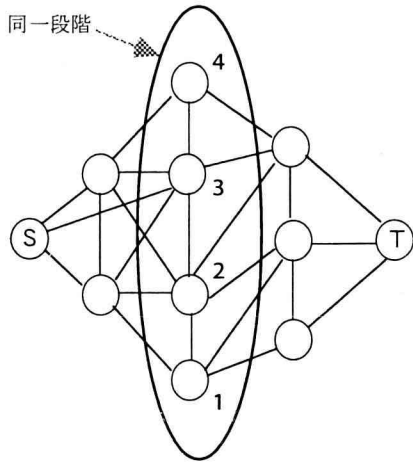


図 8: 一層複雑なネットワークの例

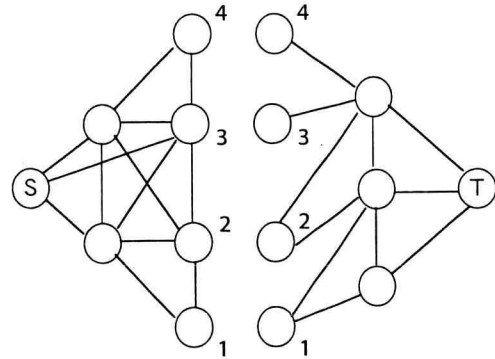


図 9: 中央の段階上での分割

うにネットワークの規模を小さく分解して、より小規模なネットワークの問題にすることが望ましい。本節では、提案アルゴリズムにより、大規模なネットワーク・アベイラビリティを求める際の分解法について述べる。

図 8は、11ノード24リンクのネットワークである [7]。このネットワークにおいてノード1, 2, 3, 4が全体を2分する一つの段階に属しているものとする。提案アルゴリズムでは、DPの最適性の原理を適用し易くするため、分解は段階上で行う (図 9)。最初に分解された右側のネットワークを対象に、図 10~13のように、それぞれノード1~4を発信ノードと見なして独立した段階として設定する。このように分解されたネットワークに関して提案アルゴリズムを適用して求めたアベイラビリティをそれぞれ A_1, A_2, A_3, A_4 とすると、もとのネットワークは図 14のような8ノード17リンクのネットワークに縮退したと見なすことができる。このような手順によって、もとネットワークと等価なネットワークに変換し、リンク数を減らすことによりアベイラビリティ計算の困難さを軽減することが可能となる。

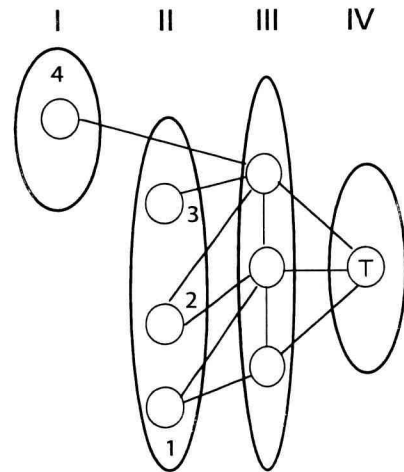


図 10: 部分ネットワークの段階 (a)

例 3 図 8に示すネットワークにおいて、各リンクのアベイラビリティをそれぞれ0.8とした場合、ここで示した方法でネットワーク・アベイラビリティを求める。はじめに縮退したアベイラビリティ A_1, A_2, A_3, A_4 を求めると次のようになる。

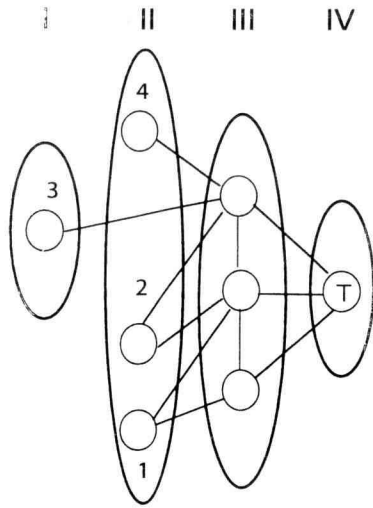


図 11: 部分ネットワークの段階 (b)

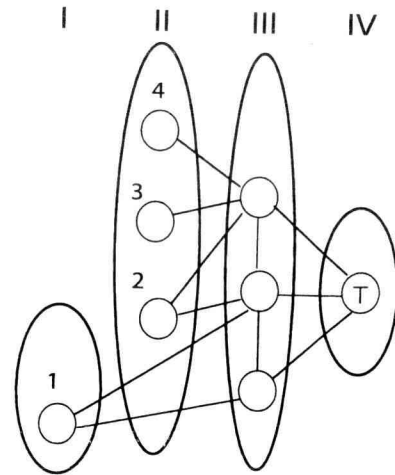


図 13: 部分ネットワークの段階 (d)

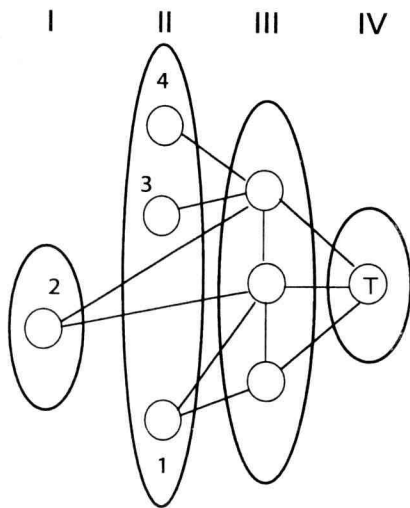


図 12: 部分ネットワークの段階 (c)

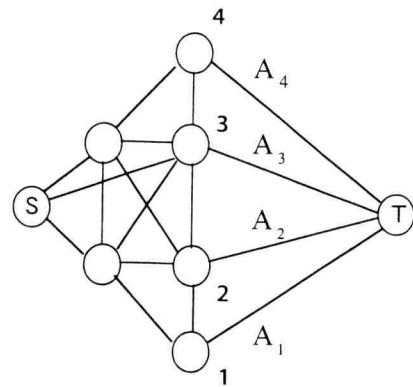


図 14: 合成された等価なネットワーク

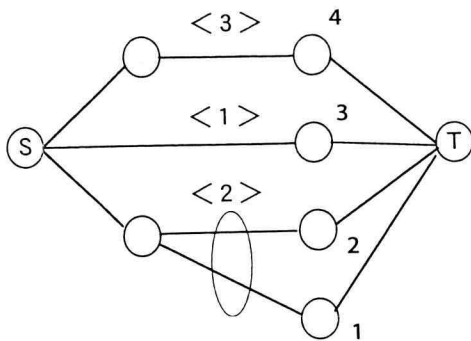


図 15: アベイラビリティの下界

A_1	:	0.9415392
A_2	:	0.9415392
A_3	:	0.7808302
A_4	:	0.7808302

また、ネットワーク・アベイラビリティは0.9884153となる。

次いでノード1, 2, 3, 4とノードT間は、縮退したアベイラビリティを用いた場合の下界について述べる。図15に示すように、DPによる最適経路を第1に取り上げる<1>。次いで第2番目と第3番目の経路を考慮したとき<2>、さらに第4番目の経路を考慮したとき<3>というやり方で下界を求める。これからアベイラビリティの下界は、それぞれ次のようになる。

<1>の場合	:	0.6246641
<1> + <2>の場合	:	0.9066481
<1> + <2> + <3>の場合	:	0.9532990

すなわち、図15に示した4経路をすべて考慮した場合、ネットワークのアベイラビリティの下界は0.953299以上と評価することができる。

7 むすび

本論文では、状態空間分解法を用いたネットワーク・アベイラビリティ計算にともなう計算困難さを改善するDPの最適性の原理にもとづいた実用的アルゴリズムの提案を行った。更に、このアルゴリズムの特徴である、より高い、ネットワーク・アベイラビリティを構成する部分を優先するという点を利用したアベイラビリティの下界計算法を示した。そして更に大規模ネットワークの分

解法および縮退法を示し、提案アルゴリズムが大規模な問題に適用可能であると考えられることを示した。

本研究は、未だ理論的検討にとどまっており、計算機実験などによる検証は、十分行われていない。今後、様々な代表的なメッシュ型ネットワークにおいて、最適解と本論文で提案した手法によって得られた解を計算機実験などにより比較し、計算時間の比較、近似解の精度、計算可能なネットワークに規模について検討を行う予定である。

謝辞 本研究を進めるにあたり、有益なご助言を頂いた本学情報工学科元教授大前義次先生に感謝致します。

参考文献

- [1] 浅谷, 通信ネットワークの品質設計, 電子情報通信学会, 1993.
- [2] 信頼性用語: J I Sハンドブック, 品質管理, p. 64, 日本規格協会, 1990.
- [3] Paul E. Renaud, Introduction to Client/Server Systems, John Wiley & Sons, 1993.
- [4] Gilbert Held, Local Area Network Performance, John Wiley & Sons, 1993.
- [5] Aaron Kershenbaum, Telecommunications Network Design Algorithms, McGraw-Hill, Inc., 1993.
- [6] 真壁肇他, オペレーションズ・リサーチ, pp. 143-148, 日本規格協会, 1980.
- [7] Berchelt, F. and Tittman, P., Reliability Analysis of Communication Networks by Decomposition, Micro-electron. Reliab, Vol. 31, No. 15, pp. 869-872, 1991.
- [8] 大前, 荒木, 小高, 家村, 通信ネットワークの信頼性解析, 信学技報, Vol. 96, No. 31, pp. 9-16, 1996.