

Title	グラフ理論的手法によるシステム信頼度解析に関する 研究
Author(s)	東山,陽一
Citation	大阪大学, 1993, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3072898
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

# グラフ理論的手法によるシステム信頼度解析に関する研究

1993年7月

東山陽一

# 内容梗概

本論文は筆者が愛媛大学(昭和52年計算機室、昭和58年情報処理センター、昭和62年工学部電気工学科、平成3年工学部電気電子工学科)ならびに平成3年度文部省内地研究員として大阪大学工学部情報システム工学科において行ってきた研究のうち、システム信頼度解析に関する研究成果をまとめたものである。

社会構造の多様化,高度化が進展するに伴い,通信網,電力網等のネットワークシステムの規模が大きくなるにつれ,ますます高い信頼性(信頼度)が要求されるようになってきた。これらシステムの信頼度が予測できれば、システムの計画、構築、保守運用の各段階において、要求された信頼度を満足しているかどうかを判定することが可能となり、またより高い信頼度を持つシステムの構築に対する考察も容易となる。

現存するシステムは多様なモデルで表現されるが、本研究では、その構成がネットワーク構造で表現可能なシステム(例えば通信網、計算機ネットワーク等)、およびその構成がネットワーク構造で表現できないシステム(例えばマルチプロセッサ・システムなどのように $\pi$ 台のプロセッサのうち少なくとも $\pi$ 台のプロセッサが同時に正常な動作をしていれば、所定の処理能力を保持し得るシステム [ $\pi$ 0 u t - o f -  $\pi$ 0 システム])を対象として、これらシステムの信頼度解析算法について考察する。

第1章緒論においては、本研究の目的ならびにその工学上の意義およびこの 分野での研究の現状について述べ、本研究の新しい諸成果について概説する.

第2章では、ネットワークの指定された2項点s, t間の信頼度(ソース・ターミナル信頼度、s-t信頼度)、すなわちソースsからターミナルtに至る正常な枝だけから成るパスが少なくとも1つ存在する確率、を求める算法について述べる。既存の算法として、与えられたネットワークからパスを開放除去してs-tアサイクリックグラフを生成するパス除去法、パス集合から得られた積項をブール演算により互いに素な積項に展開するSDP法があるが、本

章では、まずパスを順次付加することによりs-tアサイクリックグラフを生成するパス付加法を考察する、これは、

- (i) 演算時間がグラフが含む互いに枝が素なサイクルの数に大きな影響を受けない。
- (ii) 任意のレベルiでその生成を打ち切っても,i個以下のすべてのパスが組み合わされたs-tアサイクリックグラフが生成でき,信頼度の近似計算が可能である。

という特徴を持っている。さらに、グラフ理論的方法による木展開法をも提案し、この算法による信頼度表現式の算出がSDP法によるそれの算出に比べメモリおよび演算時間を大幅に減少できることを示す。また、パスの処理順序が木展開法、SDP法によって算出される積項数にどのような影響を与えるかについて述べ、計算機を用いて各算法の計算効率について比較実験を行い、改良木展開法が最良の算法であることを示す。

第3章では、ネットワークのソースとそれ以外のすべての頂点間の信頼度 (ソース・全端子信頼度、SAT信頼度), すなわちソースからソース以外の すべての頂点に至る正常な枝による経路が少なくとも一つ存在する確率,を求 める算法について論じる、ソースを根(root)とする有向木(arborescence) あるいはソースとそれ以外の頂点を分離するカットセットを求め,これにSD P法を適用して信頼度表現式を求める算法が既に提案されているが、メモリ、 演算時間に関して適切な方法とはいいがたい.本章では,まず枝の短絡,開放 を基本とした木生成法を述べ、ブール演算することなく、木の総数に等しい積 項数を持つ表現式を求める算法を示す,木生成法は木に対応するブール積項間 の比較が必要でないため高速であり、またカットセットを用いたSDP法より も使用するメモリが遙かに少ないという特徴を持っている.信頼度表現式が与 えられるとき,信頼度は各枝の信頼確率を対応するブール変数に代入すること により求められるので,計算の手間は信頼度表現式に含まれる演算記号数(リ テラル数)の増大と共に急増する.次に.アサイクリックグラフの信頼度表現 式の積項数がグラフの規模に関係なく1となることを利用したアサイクリック サブグラフ生成法,さらにサイクルを含むグラフの強連結成分を1つの頂点に

縮約して得られるグラフがアサイクリックグラフとなることを利用したセクショングラフ法について述べる。

アサイクリックサブグラフ生成法は、表現式が因数分解された形式で得られるため、表現式に含まれる演算記号数を大幅に減少できる。これに対して、セクショングラフ法で得られる表現式の積項数がさらに減少できること、および積項数が1となる条件、を示す。さらに、これら各算法に対する効率に関して比較実験を行い、セクショングラフ法が優れていることを示す。

第4章では、k-out-of-n システムの信頼度解析について論じる. 与えられたシステムにおいて、システムに含まれるn 個の要素のうち、少なくともk 個の要素が正常(異常)ならば、システムが正常(異常)となるものをk-out-of-n:G(F) システムという。 Heidtmann は、システムを正常とする要素の可能な組み合せをすべて列挙する改良 I E 法を提案している. これに対して、2章で提案した木展開法を応用することを試みる. この算法は、表現式の各積項が根付木のある種の部分木(積項木という)に1対1対応していること、積項の数が根付木の葉を除く頂点の数に等しいこと、およびスタック上の操作だけですべての積項が連続的に生成できること、という特徴を持っている. いずれの算法もn, k の増加に伴いその計算時間が指数的に増大するため、適用できるシステムの規模は小さなものに限られる.

なお、このシステムの近似信頼度を求める算法も種々提案されているが、いずれもシステム規模、計算精度、あるいは適用範囲に難点がある。そこで本章では、各要素の信頼確率が同一であるとき、その信頼度がO(k)で計算できることを活用したO(n)の近似アルゴリズムを提案する。この算法は、与えられたシステムと等しい信頼度を持ち、かつ各要素の信頼確率が同一である擬似システムにおいて、k=1、nの信頼確率から残りの信頼確率が直線近似によって推定できることを利用して、任意のkに関する信頼度を算出するものである。従来の算法に比べ提案した近似信頼度を求める算法は、

- (i) 要素数 250 程度のシステムにまで適用可能である,
- (ii) 演算時間は要素数 250 のシステムでも約 2.0 msec である, という特徴を持っている.

第5章では、本研究で得られた成果を要約し、今後に残された課題を述べる。

# 関連発表論文

# A 学会論文誌

- (A1) 東山陽一, 有吉弘: "s-tアサイクリックグラフの算出アルゴリズム", 電子通信学会論文誌(A), Vol. J67-A, No. 4, pp. 367-374(1984-4).
- (A2) 東山陽一, 有吉弘: "木展開法による新しいネットワーク信頼度計算法", 電子通信学会論文誌(A), Vol. J68-A, No. 2, pp. 122-129(1985-2).
- (A3) 東山陽一, 有吉弘: "k-out-of-n システムへの木展開法の応用", 電子通信学会論文誌(A), Vol. J68-A, No. 6, pp. 594-601(1985-6).
- (A4) 東山陽一,有吉弘: "信頼度表現式の積項数削減の一手法",電子通信 学会論文誌(A), Vol. J69-A, No. 5, pp. 560-566(1986-5).
- (A5) 東山陽一,渡敏広,有吉弘: "k-out-of-n システムの近似信頼度算出アルゴリズム",電子情報通信学会論文誌(A), Vol. J70-A, No. 2, pp. 212 -219(1987-2).
- (A6) 野田隆志,東山陽一,有吉弘,白川功: 木を用いたソース・全端子信頼度表現式の最適生成法",電子情報通信学会論文誌(A), Vol. J72-A,No. 2, pp. 381-389(1989-2).

### B 国際会議

- (B1) H. Ariyoshi and Y. Higashiyama: "New method for computing exact terminal reliability using a tree expansion technique", Proceedings of 1984 International Symposium on Circuits and Systems, pp. 1135-1138 (1984-5).
- (B2) Y. Higashiyama and H. Ariyoshi: Tree-expansion technique applied to k-out-of-n systems, Proceedings of 1985 International Symposium on Circuits and Systems, pp. 1651-1654(1985-6).
- (B3) Y. Higashiyama, T. Watari, and H. Ariyoshi: "An algorithm for computing approximate reliability of k-out-of-n systems".

- Proceedings of the 25th SICE Annual Conference, pp. 1059-1062 (1986-7).
- (B4) Y. Higashiyama, T. Noda, H. Ariyoshi, and I. Shirakawa: "A new topological method for symbolic reliability analysis of source-to-all-terminal networks", Proceedings of the 27th SICE Annual Conference, pp. 911-914(1988-8).
- (B5) Y. Higashiyama, T. Noda, H. Ariyoshi, and I. Shirakawa: "Minimizing symbolic reliability formula of source-to-all-terminal networks", Proceedings of 1988 Joint Technical Conference on Circuits/
  Systems, Computers and Communications, pp. 68-72(1988-11).
- (B6) Y. Higashiyama, H. Ariyoshi, and I. Shirakawa: "An algorithm for generating a factored symbolic reliability formula of directed source-to-all-terminal network", Proceedings of 1989
  International Conference on Circuits and Systems, pp. 774-777
  (1989-7).
- (B7) Y. Higashiyama, H. Ariyoshi, I. Shirakawa, and S. Ohba: "A method for generating a factored SAT symbolic reliability formula", Proceedings of 1990 Joint Technical Conference on Circuits/Systems, Computers and Communications, pp. 333-337(1990-12).

## C 学会研究会

- (C1) 東山陽一, 有吉弘:"s-tアサイクリック部分グラフの算出アルゴリズム", 電子通信学会技術研究報告, CAS82-77, pp. 53-60(1982-10).
- (C2) 東山陽一, 有吉弘: "木展開法による新しいネットワーク信頼度計算法", 電子通信学会技術研究報告, CAS83-113, pp. 23-28(1983-9).
- (C3) 有吉弘,東山陽一,: "ネットワーク信頼度評価のグラフ論的手法", 電子通信学会技術研究報告,IT83-31,pp.7-12(1983-11).
- (C4) 東山陽一,渡敏広,有吉弘:"k-out-of-n システムへの木展開法の応用 ",電子通信学会技術研究報告,CAS84-72,pp.79-85(1984-8).

- (C5) 東山陽一,向井昌規,有吉弘:"信頼度表現式の積項数削減の一手法", 電子通信学会技術研究報告,CAS85-19,pp.53-59(1985-7).
- (C6) 東山陽一,渡敏広,有吉弘: "k-out-of-n システムの近似信頼度の一計算法",電子情報通信学会技術研究報告,CAS85-136,pp.33-40(1986-1).
- (C7) 東山陽一,野田隆志,有吉弘,白川功:"積項生成におけるカットセットの処理順序",電子情報通信学会技術研究報告,CAS86-179,pp. 39-45(1987-1)。
- (C8) Y. Higashiyama, H. Ariyoshi, and I. Shirakawa: "New algorithms for computing the reliability of consecutive-k-out-of-n: F systems", 電子情報通信学会技術研究報告, CAS87-90, pp. 231-236(1987-7).
- (C9) 野田隆志,東山陽一,有吉弘,白川功: \*\* 木を用いたソース・全端子信頼度表現式の最適生成法\*\*,電子情報通信学会研究技術報告,CAS87-205,pp.55-60(1987-12).

# D 紀要

- (D1) 東山陽一, 有吉弘: "ネットワーク信頼度評価のグラフ論的手法", 愛媛大学工学部紀要, Vol. 11, No. 1, pp. 43-53(1986-2).
- (D2) 東山陽一,有吉弘,白川功:"積項生成におけるカットセットの処理順序と演算量の軽減効果",愛媛大学工学部紀要,Vol.11, No.4, pp.95-103(1989-2).
- (D3) Y. Higashiyama, T. Noda, H. Ariyoshi, and I. Shirakawa: "An optimal symbolic reliability formula using spanning trees for source-to-all-terminal systems", 愛媛大学工学部紀要, Vol. 12, No. 1, pp. 117-132(1990-2).

# グラフ理論的手法によるシステム信頼度解析に関する研究

# 目次

第1	章	Ì	緒	論	i	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		1
第2	章	ì	ネ	ッ	<b>١</b>	ワー		ク・	の	ソ	_	ス	•	タ	_	3	ナ	ル	信	頼	叓		•	•	•	•	•	•	•	•		5
2		1		緒	言		•		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		5
2		2		諸	定	義	b.	ょ	<b>V</b>	従:	来	の	手	法	と	既	知	の	定	理		•	•	•	•	•	•	•	•	•		7
2		3		T	RŽ	去		•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1	3
	2		3		1	,	۱۴:	ス	狳	去	法		•	•	•	•	•	•	•	•	•		•	•	•	•	•	•		•	1	4
	2		3		2	į	۱۴.	スイ	付	加	法		•	•	•	•	•	•		•	•		•	•	•	•	•		•	•	1	8
2		4		木	展	荆	去		•		•	•	•		•	•				•	•	•	•	•	•	•	•	•	•		2	3
	2		4		1	į	基	隼	木	ځ	積	項	木		•	•					•	•	•	•	•		•	•	•	•	2	3
	2		4		2	į	基	隼	木	の	縮	約	の	グ	゙ヺ	フ	理	論	的	条(	件		•		•	•	•	•	•	•	2	6
	2		4		3	1	漬	項	木	の:	生	成		•						•	•	•	•		•	•	•	•	•	•	2	9
2		5		۶۲	ス(	のり	処:	哩	順	序	ح	積	項	数		•			•	•		•	•		•	•	•	•	•	•	3	1
2		6		実	験	結	果	ع	考	察								•	•			•			•	•		•		•	3	5
	2		6		1	,	٧,	ス	除	去	法	بح	۶٧	ス	付	加	法	の	比	較		•				•			•	•	3	9
	2		6		2	. 5	木	展	開	法	ح	S	D	P	法	の	比	較		•											3	9
	2		6		3	,	۶٧	ス・	の	処	理	順	序	بح	積	項	数	の	関	係				•					•	•	4	2
2	<b>.</b>	7		結	言												•	•	•	•					•					•	4	2
第3	章	ī	ネ			ヮ・	_	ク	の	ソ	_	ス		全	端	子	信	頼	度	:											4	5
	3.				言																•			•							4	5
3	3.	2			定	義						•		•		•						•									4	7
					生		决					•																•			4	9
									度	表	現	左			•																4	9
					2					の															•		•				5	0
					3					成	-	•	•	•					•									•				2
	٠	•	•	•	-	•					•																				-	

	3	•	4		ア	サー	1	ク	リ	ッ	ク	サ	ブ	グ	゙ヺ	フ	生	成	法		•	•	•	•	•	•	•	•	•	•	•		5	9
		3		4		1	ļ	既	知	の	結	果			•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•		5	9
		3		4		2		ア	サ.	1	ク	ij	ッ	ク	サ	ブ	グ	ラ	フ	の	生	成		•	•	•			•	•	•		6	4
	3		5		積	項数	数	の:	最	小	化		•								•	•			•		•		•	•	•		7	1
	3		6		実	験網	洁	果	ځ	考	察		•			•					•	•		•	•	•			•	•			7	6
		3		6		1		木	生儿	戉	法		•	•	•			•	•		•	•	•		•	•			•	•	•		7	8
		3		6		2		ア	サ	1	ク	IJ	ッ	ク	サ	ブ	グ	ラ	フ	法		•	•	•	•				•	•	•		8	1
		3		6		3		乜	ク	シ	3	ン	グ	ラ	フ	法		. •			•	•		•	•	•			•	•	•		8	1
	3		7		結	言			•			•	•					•	•		•	•	•	•	•	•			•	•			8	3
第	4	章		k	_	<b>o</b> 1	u	t	_	o	f	_	П	シ	ス	テ	ム	信	頼	度	:	•	•	•	•			•	•	•	•		8	5
	4		1		緒	言			•		•	•		•					•	•	•	•		•	•	•		•	•	•	•		8	5
	4		2		諸	定	養		•		•	•			•	•				•	•	•		•	•	•				•	•		8	5
	4		3		木	展	荆	法	の	Ċ	用		•	•								•	•	•	•	•	•		•		•		8	7
		4		3		1		既	知	<u>の</u>	定	理	ح	本	手	法	ح	の	関	係		•	•	•	•	•	•		•	•	•		8	9
		4		3		2		, G	, , ,	Š	/ 7	くゔ	- 1	<b>L</b> 0	D ,	Вι	の		般	形		•		•	•					•			9	6
		4		3		3		"F	7 k	È	/ 7	くゔ	- 1	<b>L</b> 0	D ,	B <sub>R</sub>	の	_	般	形					•				•	•	•		9	9
		4		3		4	]	He:	idí	t m	an:	n	の	改	良	I	E	法	ح	の	比	較			•				•			1	0	0
	4		4		近	似化	言	頼	度							•		•		•		•		•	•		•	•	•		•	1	0	3
		4		4		1		木	展	荆	法	に	ょ	る	積	項	数	بح	時	間	複	雑	度		•		•	•	•	•		1	0	3
		4		4		2	1	近位	似	<b>:</b>	算	法								•		•				•			•		•	1	0	8
		4		4		3	i	誤	差(	の	評	価				•		•		•		•										1	1	3
		4		4		4	j	近	似	7	ル	ゴ	ij	ズ	<b>'</b> _	の	適	用	範	囲					•						•	1	1	4
	4		5		結	言																										1	1	9
第	5	章			論										•				•	•										•		1	2	1
謝					•							•							•					•	•									
			献	:																														

# 第1章 緒論

産業,経済,社会活動の多様化,高度化に伴って,各種のネットワークシステムがますます大規模なものとなってくるが,その典型的な例として通信網,電力網,計算機ネットワーク等があげられる。今日の社会はそれらシステムが正常に機能することを前提に活動を行っており、システムに障害が発生すると社会活動は大幅に制限される。したがって、これらシステムの信頼性を高めることは今日の情報化社会の発展にとって重要な鍵であるといえる。

システムの信頼性を「システムが所定の動作条件のもとで、所定の時間、所定の機能を継続的に果たす確率」であると定義し、これを信頼度(reliability)と名づける。信頼性の高いシステムを実現するための方法はきわめて多岐にわたっているが、これを分類すればシステムを構成する要素あるいはサブシステム自体の信頼性を向上するための手法、および与えられた要素あるいはサブシステムを用いて信頼性の高いシステムを構成するための手法、の2つに大別することができる。システム構成に冗長性を導入して耐故障性を高めること、あるいは故障の際のシステム機能の維持をはかることは後者に属している。本論文は後者の観点からシステム信頼度解析について考察を行うものである。

システムは一般にいくつかのサブシステムから成り、そのサブシステムはまたより小さなサブシステムから成つている。過去の使用実績等から故障率が解明可能な単位までシステムを分解すれば、その分解能に応じて信頼度を予測することができる。信頼度の予測(reliability prediction)が的確に行われる場合には、システムの構成にあたって、信頼性の面から要求された信頼度を満足する計画を選定することが可能となるばかりでなく、システムを構成するサブシステム間で信頼度を配分することも可能となり、さらに保守運用面の計画の立案にも資することができる。

与えられたシステムの信頼度を評価しようとするとき,それを全体として取り扱うよりもいくつかの機能的要素から構成されているものとして取り扱う方が容易である.システムの各構成要素にはいくつかの状態が考えられるが,次

の制約のもとでシステムの信頼度評価をおこなうことが多い[12].

- a. 各構成要素は2つの状態を持つ.正常(successful)と異常(fault).
- b. 各構成要素が正常である確率(信頼確率)はそれぞれ異なる. また, 異常である確率(非信頼確率)と信頼確率の和は 1.0 である.
- c. ある構成要素が異常となっても、他の要素の信頼確率に影響を与えることはない(統計的独立).

与えられたシステムの極小なパスセットあるいはカットセットを求め、それら極小集合を構成する要素に対応する信頼確率を代入しその和を求めても、システム信頼度(あるいは非信頼度)とは一致しないのが一般的である。その理由は極小集合の生起が互いに排他的でないためである。そこで、本論文では信頼確率を代入することにより信頼度を直接算出できる各種算法を考察する。与えられた要素を用いて最適なシステムを設計する問題、あるいは要素の信頼確率の摂動に伴うシステム信頼度の変動を問題にする感度問題があるが、これらの問題を解決するには信頼度および感度を繰り返し計算する必要があり、信頼度のブール表現式を導いておくことは極めて重要であると考えられる。このことから、本論文では信頼度計算に1対1対応する信頼度表現式を導くことに主眼をおく。

現存するシステムは多様なモデルで表現されるが、本論文ではその構成がネットワーク構造で表現可能なシステム[1](例えば通信網、計算機ネットワーク等)、およびその構成がネットワーク構造で表現できないシステム[1](例えばマルチプロセッサ・システムなどのように $\pi$ 台のプロセッサのうち $\pi$ 台のプロセッサが同時に正常な動作をしていれば、所定の処理能力を保持し得るシステム [ $\pi$ 0 u t - o f -  $\pi$ 0 システム])を対象として、これらシステムの信頼度解析算法について考察する。

第2章では、ネットワークの指定された 2 頂点 s 、 t 間の信頼度(ソース・ターミナル信頼度、s-t 信頼度)、すなわちソース s からターミナル t に至る正常な枝だけから成るパスが少なくとも 1 つ存在する確率、を求める算法について述べる。既存の算法として、与えられたネットワークからパスを開放除

去してs-t アサイクリックグラフを生成するパス除去法[2],パス集合から得られた積項をブール演算により互いに素な積項に展開するSDP法[15]がある。これに対して、パスを順次付加することによりs-t アサイクリックグラフを生成するパス付加法[4,11]、およびグラフ理論的方法による木展開法[8,17,18,20,21]を提案し、さらにパスの処理順序が木展開法、SDP法によって算出される積項数にどのような影響を与えるかについて述べる[22,23]。

第3章では、ネットワークのソースとそれ以外のすべての頂点間の信頼度(ソース・全端子信頼度、SAT信頼度)、すなわちソースからソース以外のすべての頂点に至る正常な枝による経路が少なくとも一つ存在する確率、を求める算法について論じる、ソースを根(root)とする有向木(arborescence)あるいはソースとそれ以外の頂点を分離するカットセットを求め、これらにSDP法を適用して信頼度表現式を求める算法が既に提案されているが[24]、メモリ、演算時間に関して適切な方法とはいいがたい。本章では、まず枝の短絡、開放を基本とした木生成法を述べ、ブール演算することなく、木の総数に等しい積項数を持つ表現式を求める算法[39,40,41,42,43]を示す。さらに、アサイクリックなグラフの信頼度表現式の積項数がグラフの規模に関係なく1となること[44]を利用したアサイクリックサブグラフ生成法[45]、さらにサイクルを含むグラフの強連結成分を1つの頂点に縮約して得られるグラフがアサイクリックグラフとなることを利用したセクショングラフ法[46]について述べる。

第4章では、k-out-of-n システムの信頼度について論じる. 与えられたシステムにおいて、システムに含まれる $\pi$ 個の要素のうち、少なくともk個の要素が正常(異常)ならば、システムが正常(異常)となるものをk-out-of-n:G(F) システムという。 Heidtmann [48]は、システムを正常とする要素の可能な組み合せをすべて列挙する改良 I E 法を提案しているが、これに対して、2章で提案した木展開法を応用することを試みる[50,52,53].

信頼度を求めるいずれの算法も n, k の増加に伴いその計算時間が指数的に 増大するため、適用できるシステムの規模は小さなものに限られる。そのため、 システムの近似信頼度を求める算法も種々提案されているが、いずれもシステ ム規模, 計算精度, あるいは適用範囲に難点がある[1,48]. そこで, 与えられたシステムと等しい信頼度を持ち, かつ各要素の信頼確率が同一である擬似システムにおいて, k=1, nの信頼確率から残りの信頼確率が直線近似によって推定できることを利用して, 任意のkに関する信頼度を算出する方法を提案する[55,56,57].

第5章では、本研究で得られた成果を要約し、今後に残された課題を述べる.

# 第2章 ネットワークのソース・ターミナル信頼度

## 2.1 緒言

おける基本的問題の1つである[1].

複雑なシステムの信頼度を評価しようとするとき、それを全体として取り扱うよりもシステムをいくつかの機能的成分に分けて取り扱う方が容易である。各成分にはいくつかの状態が考えられるが、ここではネットワーク構造を持つシステムに限定し、各構成要素が2つの状態—正常(successful)、異常(fault)—を持つものと考える。このように限定すると、システムは各構成要素に信頼確率(正常な確率)あるいは非信頼確率(異常な確率)が付与された重み付きグラフ(信頼性グラフ(reliability graph))として表現できる[1]。システムのある成分が異常となってもその他の成分の信頼度に影響を与えない(統計的独立)という制約のもとで、指定された2頂点(s, t)に関してシステムが正常である確率は、システムに対応するグラフ上において、s から

与えられた要素を用いて最適なシステムを設計する問題,あるいは要素の信頼確率の摂動に伴うシステム信頼度の変動を解析する感度問題においては,信頼度および感度を繰り返し計算する必要があり,信頼度のブール表現を導いておくことは極めて重要と考えられる.

t に至る正常な枝だけから構成されるs-t パスが少なくとも1 つ存在する確率を意味している。このような確率を求めることは、システムの信頼度解析に

道路網に対しては,一方通行の道路を有向枝で,双方向通行路を互いに逆向きの2本の有向枝で,また道路の交差点を頂点で表して,一つの有向グラフとして描かれるが,2章で取り扱う信頼性グラフは,このような有向グラフの枝に信頼確率pあるいは非信頼確率q (=1-p)を与え,頂点は常に正常(successful)であるものとする。なお,頂点に信頼確率が与えられるときには,頂点を頂点と等しい信頼確率を持つ有向枝で置き換えるものとする。また,与えられたグラフが無向枝を含むとき,この無向枝を互いに逆向きの2本の有向

枝に置き換え、その信頼確率は無向枝の信頼確率を持つものとして取り扱う。 システムの信頼度を求める算法として、すべてのs-tパス  $\{\mu_1, \mu_2, \cdots, \mu_m\}$  の列挙を基本操作とする以下の算法がある。

- (i) 直接展開法[1]
- (ii) 繰返しIE法 (inclusion-exclusion method) [9]
- (iii) TR法 (topological reliability method) (あるいはs-tアサイクリックグラフ生成法) [2]
- (iv) 簡易展開法[12]
- (V) SDP法 (sum of disjoint products method) [13,14,15] また、カットセットの発生を基本操作とする以下の算法がある.
  - (vi) 直接展開法[1]
  - (vii) SDP法[13.14.15]
  - (viii) 修正カットセット法[16]
- (i) および (vi) は2 <sup>11</sup>−1 個の組み合せを考える必要があり、また各組み 合せは補元 (complement) を用いないため、得られるブール表現には積項( disjoint product) の論理和および論理減算を含み、互いに相殺可能な積項が 数多く存在する[2]. (i)における互いに相殺されない積項だけを求める方法 として(ii), (iii)がある. (ii)はブール演算による方法, (iii)はグラ フ理論的方法であり、両者とも積項数をかなり減少させることができるが、グ ラフの規模の増大とともに積項数は指数的に増加する. (iv) は,公式では積 項数が2 <sup>m-1</sup>と(i)の半分となるが、実際のグラフに適用した場合、多くの積 項が抹消され、積項数を大幅に減少させることができる[12].しかしながら、 積項の展開において,2´´´ー ェー1個もの展開公式の記憶を必要とする.(v), (vii)は,パスあるいはカットセットから得られる積項をブール演算により互 いに素な積項の和に展開するものであるが、演算の過程で生じる積項をすべて 記憶する必要があり、また演算時間が展開のための比較回数によって大きく支 配される等の欠点を持っている.(viii)は小規模なグラフに限り有効である. また、Locks は繰返しIE法およびSDP法に関する比較検討を行い、SDP 法の優位性を指摘している[9.10].

本章ではネットワークのs-t 信頼度表現式を求める方法を4節に分けて述べる。2.2節では諸定義と既に得られている成果を、2.3節ではパスを順次付加することによりs-t アサイクリックグラフを生成するT R法[4,11]を、2.4節ではグラフ理論的方法により互いに素な積項を求める木展開法[8,20,21]を、2.5節ではパスの処理順序が得られる積項数にどのような影響を与えるかを調べ、それを基に改良木展開法[22,23]を提案する。また、提案した算法と既存の算法に対して、計算機実験による比較検討を行う。

# 2.2 諸定義および既存の方法と既知の定理

有向グラフを G = (V, E) で表す。ここで、Vは頂点の集合、Eは有向枝の集合である。枝  $e \in E$ に対して、その枝の始点および終点を、それぞれ S (e) 、 t (e) で表す。

互いに相異なる頂点と枝の列  $\{v_0, e_1, v_1, e_2, v_2, \cdots, e_{dis}, v_{dis}\}$  が  $s(e_i) = v_{i-1}, t(e_i) = v_i (1 \le i \le dis)$  の関係を満たすとき、これを  $v_0$ から  $v_{dis}$ への長さ dis のパスといい、 $\mu(v_0, v_{dis})$  と書く、特に、 $v_0 = s$ 、 $v_{dis} = t$  のとき、s - t パスという。また、 $v_0 = v_{dis}$  の場合、長さ dis のサイクルという。いくつかのサイクルの集合において、どの2つのサイクルも枝を互いに共有しないとき、これらのサイクルを互いに枝が素なサイクルという。 G の頂点  $v \in V$  に対して、s(e) = v とする枝 e の個数および t(e) = v とする枝 e の個数を、それぞれ v の正次数、負次数といい、それぞれ  $d^+(v)$  および  $d^-(v)$  と書く。 $d^-(\cdot) = 0$  である頂点および  $d^+(\cdot) = 0$  である頂点を、それぞれソース、g ーミナルと呼ぶ。

[定義2-1] 部分グラフ $G_i \subseteq G$ がそれぞれただ1個のソースsおよびターミナルtを含み,かつ $G_i$ のどの枝も少なくとも1つのs-tパス上の枝となり得るとき, $G_i$ をs-tグラフという.また,サイクルを含むs-tグラフおよび全く含まないs-tグラフを,それぞれs-tサイクリックグラフおよびs-tアサイクリックグラフといい, $G_s$ および $G_s$ と書く.

例えば、図2-1のグラフGはs-tサイクリックグラフ、図2-1から枝gを開

放除去したグラフ(図2-3の下段右端のグラフ)はs-t アサイクリックグラフ である.

[定義2-2]  $G_i \subseteq G$ において、ある枝がどのs = tパス上にもないとき、この枝をs = tパス孤立枝という。

[定義 2-3] s-t アサイクリックグラフ  $G_s$ において、パス $\mu$ ( $v_0$ 、 $v_{dis}$ )が

- (i)  $v_0 = s$ ,  $\delta v_0 \neq s$   $\delta v_0 \neq s$
- (ii)  $v_{dis} = t$ ,  $bar{s} = t \lambda bil d^{-}(v_{dis}) > 1$
- (iii)  $\mu$  ( $v_0$ ,  $v_{dis}$ ) 上のどの中間頂点  $v_i$  ( $1 \le i \le dis 1$ ) も  $d^+$  ( $v_i$ ) =  $d^-$  ( $v_i$ ) = 1

の関係を満たすとき、このパスを中立パスと呼ぶ.

例えば、図2-3下段右端のグラフにおいて、枝b、dで構成されるパスは中立パスの1つである。

また、記述を簡単するため以下の記号を定める.

 $n_{G}$ ,  $l_{G}$ , m: グラフ G の頂点数、枝数、s-t パス数.

 $x_e(\bar{x}_e)$ : 枝 e が正常 (異常) であることを示すブール変数.

 $p_e(q_e)$ :枝 e が正常(異常)である確率.

 $\mu_i$ : s-tパスを構成する枝および頂点の順序列,またはs-tパスの枝集合.

 $A_i: s-t$ パス $\mu_i$ に属する枝eに対応したブール変数 $x_e$ の積.

Pr {A<sub>i</sub>} : A<sub>i</sub>の信頼度.

R: Gの信頼度.

 $F_R$ : Gの信頼度のブール表現.

 $|F_R|:F_R$ の積項数.

グラフGにおいて、そのs-tパスはGの深さ優先探索(depth-first search、DFS)により効率よく生成できる、22-101-5間の6個のパス

 $\mu_1 = \{1, a, 3, f, 5\}$ 

 $\mu_2 = \{1, a, 3, e, 4, h, 5\}$ 

 $\mu_3 = \{1, b, 2, d, 3, f, 5\}$ 

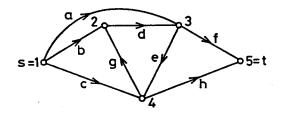


図2-1 グラフ G

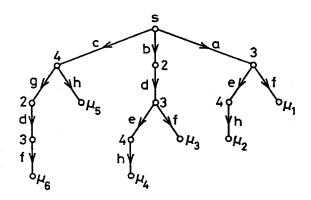


図2-2 Gの有向木 T

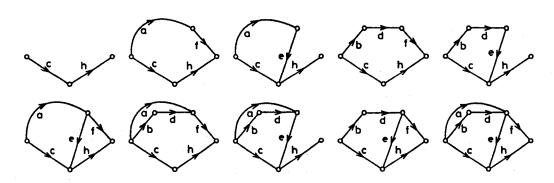


図2-3 s-tアサイクリックグラフの集合  $g_s$ 

$$\mu_{4} = \{1, b, 2, d, 3, e, 4, h, 5\}$$

$$\mu_{5} = \{1, c, 4, h, 5\}$$

$$\mu_{6} = \{1, c, 4, g, 2, d, 3, f, 5\}$$

は、その生成順に従い、図2-2のように有向木 Tで表すことができる。 Tの頂点 および枝は、グラフ G上の頂点および枝に対応し、それぞれ Gの頂点、枝に相 当するラベルを持つ。

以下では、根付木における子、親、子孫、祖先、兄弟、葉、根、などの用語は従来の慣例に従うものとする[30].

与えられたグラフGにおいて、s、t間の信頼度はsからtに至る正常な枝だけから成るパスが少なくとも1つ存在する確率を意味することから、s-tパス $\mu_i$ (i=1, 2, …, m)に対応するブール変数の積 $A_i$ を用いて信頼度表現式 $F_s$ が次式で書ける[1].

$$F_R = A_1 + A_2 + \cdots + A_m \tag{2-1}$$

式(2-1)の $A_i$  (i=1, 2, …, m) が一般には互いに素 (disjoint,  $A_i$ ・ $A_i=0$ ) でないため、実際には次のような展開定理 (直接展開法) により信頼 度 R が求められる[1].

$$R = P r \{A_1\} + P r \{A_2\} + \cdots + P r \{A_m\}$$

$$- P r \{A_1 A_2\} - P r \{A_1 A_3\} - \cdots - P r \{A_{m-1} A_m\}$$

$$+ P r \{A_1 A_2 A_3\} + P r \{A_1 A_2 A_4\}$$

$$+ \cdots + P r \{A_{m-2} A_{m-1} A_m\}$$

$$\vdots$$

 $+ (-1)^{m} P r \{A_{1} A_{2} \cdots A_{m}\}$ 

ここで、 $A_1A_2$  ···  $A_i$ は $\mu_1$ 、 $\mu_2$ 、··· ,  $\mu_i$ が同時に正常であることを示すブール変数の積である.

(2-2)

図2-1のs-tパスを、式(2-2)で展開すれば表2-1が得られる。式(2-2)のj行は表2-1のj列に対応しており、またj行の符号(sign)を表2-1の各列の上部に sign として示している。表2-1の(i, j)ブロック内にある各積項は  $\{A_1, A_2, \cdots, A_k, \cdots, A_{j-1}\}$  から選ばれたj-1 個の $A_k$ と $A_j$ の積から作られている。また、表2-1には $A_1A_4=x_bx_dx_ax_bx_ax_f$ 、 $A_1A_2A_4=$ 

表2-1 直接展開法

sign	+	_	+	-	+
<i>j</i> =1	$A_1 = X_a X_f$				
<i>j=</i> 2	$A_2 = X_2 X_2 X_h$	$A_1 A_2 = X_0 X_0 X_h X_f$			
<i>j</i> =3	A3=X6X6X7	$A_1 A_8 = X_b X_d X_f X_\theta$ $A_2 A_8 = X_b X_d X_f X_b X_b X_b$	A1 A2 A3 = X 5 X 0 X 7 X 0 X 0 X h		
<i>j=</i> 4	A4=XbXdXeXh	A1 A1 A2 A1 A8 A1	= Ab X a X b X b X b X b X b X a X b X b X	A1 A2 As A4 = Xb X d Xe Xb Xo Xf	
7=5	As = Xc Xn	$A_1 A_5 = X_C X_h X_\theta X_f$ $A_2 A_5 = X_C X_h X_\theta X_\theta$ $A_3 A_5 = X_C X_h X_b X_d X_f$ $A_4 A_5 = X_C X_h X_b X_d X_\theta$	A1 A2 A8 = XC XA X4 X7 X8 A1 A3 A8 = XC XA X4 X7 X8 X1 A1 A4 A8 = XC XA X4 X7 X8 X1 X8 A2 A3 A8 = XC XA X4 X4 X8 X4 X4 A2 A4 A8 = XC XA X4 X6 X6 X6 X6 A3 A4 A8 = XC XA X6 X6 X6 X6 A3 A4 A8 = XC XA X6 X6 X7 X6	A1 A2 A6 = XcXhXeXfXe       A1 A2 A4 A6 = XcXhXeXfxXfXeXbXd         A1 A3 A6 = XcXhXeXfXf       A1 A2 A4 A6 = XcXhXeXfXfXd         A1 A4 A6 = XcXhXeXfXfXfXf       A1 A3 A4 A6 = XcXhXeXfXfXfXf         A2 A4 A6 = XcXhXaXeXbXd       A2 A3 A4 A6 = XcXhXaXeXbXd         A3 A4 A6 = XcXhXbXdXfXf       A3 A3 A4 A6 = XcXhXbXdXfXf	A1 A2 A8 A6 = Xc Xh Xe Xf Xe Xb Xd     A1 A2 A8 A4 A6 = Xc Xh Xa Xf Xe Xb Xd       A1 A2 A4 A6 = Xc Xh Xa Xf Xe Xb Xd     A1 A8 A4 A6 = Xc Xh Xa Xf Xb Xd Xf       A2 A8 A4 A6 = Xc Xh Xa Xe Xb Xd Xf     A2 A8 A4 A6 = Xc Xh Xa Xe Xb Xd Xf
<i>j=</i> 9	A8 = Xc Xg Xd Xf	•	•	•	* • • • • • • • • • • • • • • • • • • •

 $x_0 x_0 x_0 x_0 x_0 x_0 x_0 x_0$ のように異符号で同一のブール表現となる項が存在するため、互いに相殺される項が含まれている(下線で示す).

直接展開法において,互いに相殺される積項を展開の途中で,あるいはあらかじめ除外して行う算法として繰返し I E 法 [9] がある.表2-1において,i=3,4の展開の途中で下線部の積項が相殺され,以後の演算から除外される.i=5 では, $i=1\sim4$  の各積項(相殺されない)を用いて, $2\sim5$  列をそれぞれ導く.その結果,下線部を除外した積項が得られる.

表2-1における相殺されない積項が1 つのs - t アサイクリックグラフに対応することに注目し、グラフ理論的方法で順次s - t アサイクリックグラフを発生する算法がある[2,4,11]. i 番目のs - t パスに関するs - t アサイクリックグラフの集合(例えば、表2-1のi行)をg で、g が寄与する信頼度をP (g) で表せば、信頼度R が次式のようにも表現できる[4,11].

$$R = P(\mathcal{G}_1) + P(\mathcal{G}_2) + \cdots + P(\mathcal{G}_m)$$
 (2-3)  
ここで、 $\mathcal{G}_1$ はパス $\mu_1$ そのものであることに注意しなければならない(表2-1の  
1行参照).

表2-1の i=5 における s-t アサイクリックグラフの集合  $\mathcal{G}_s$ を図2-3に示す。図2-3から、 $P(\mathcal{G}_s)$  が次式のように得られる。

$$P (\mathcal{J}_{6}) = p_{ch} - p_{ch} (p_{af} + p_{ae} + p_{bdf} + p_{bde}) + p_{ch} (p_{afe} + p_{afbd} + p_{aebd} + p_{bdfe}) - p_{ch} \cdot p_{afebd}$$
(2-4)

ここで、 $p_X = \prod_{X \in X} p_X$ とする.

また,式(2-4)は次のように書くことができる.

$$P (\mathcal{G}_{5}) = P_{ch} (1 - P_{r} \{X_{s}X_{f} + X_{s}X_{e} + X_{b}X_{d}X_{f} + X_{b}X_{d}X_{e}\})$$
(2-5)

このとき,枝列 a f , a e , b d f , b d e のそれぞれは, $\mu$   $_{5}$ 上に始点および終点を持ち, $\mu$   $_{5}$ と互いに素な枝から構成される $\mu$   $_{1}$  ,  $\mu$   $_{2}$  ,  $\mu$   $_{3}$  ,  $\mu$   $_{4}$  の部分パスとなっている.この部分パスを $\mu$   $_{15}$  ,  $\mu$   $_{25}$  ,  $\mu$   $_{35}$  ,  $\mu$   $_{45}$  , そのブール変数の積 A  $_{15}$  , A  $_{25}$  , A  $_{35}$  , A  $_{45}$  で表せば,式(2-5)は次のように表現できる.

$$P (\mathcal{G}_{5}) = P_{ch} (1 - P_{r} \{A_{15} + A_{25} + A_{35} + A_{45}\})$$
 (2-6)

 $P(\mathcal{G}_{\mathbf{5}})$  のブール表現を $F_{\mathbf{5}}$ で表せば、式(2-6)は次式のように書ける.

$$F_5 = A_5 \{ 1 - (A_{15} + A_{25} + A_{35} + A_{45}) \}$$

$$= A_{5} \left( \overline{A_{15} + A_{25} + A_{35} + A_{45}} \right)$$

$$= \overline{A_{15}} \, \overline{A_{25}} \, \overline{A_{35}} \, \overline{A_{45}} \, A_{5}$$
(2-7)

式(2-7)の表現は木展開法の基礎となるものである[8,20,21].

一方, SDP法に関する展開公式として次式がある[9].

$$F_R = A_1 + \overline{A_1} A_2 + \overline{A_1} \overline{A_2} A_3 + \cdots + \overline{A_1} \overline{A_2} \cdots \overline{A_{m-1}} A_m$$
 (2-8)  
ここで、i番目の項 $\overline{A_1} \overline{A_2} \cdots \overline{A_j} \cdots \overline{A_{j-1}} A_j \circ \overline{A_j} \otimes \overline{A_$ 

$$\bar{A}_{i} = \bar{X}_{1} + X_{1} \bar{X}_{2} + X_{1} X_{2} \bar{X}_{3} + \cdots + X_{1} X_{2} \cdots X_{k-1} \bar{X}_{k}$$
 (2-9)

SDP法のステップiでは、互いに素な積項の集合をPDで表せば、

 $PD_1 = \overline{A_1}A_i$ ,  $PD_2 = \overline{A_2}PD_1$ , …,  $PD_{i-1} = \overline{A_{i-1}}PD_{i-2}$  (2-10) の順に繰返し展開される[15]. このとき,  $PD_i$ から $PD_{i+1}$ の生成には,  $PD_i$ の各項と $A_{i+1}$ の比較が行われる.

### 2.3 TR法

従来,正確な信頼度を求める算法として,s-tカットセットあるいはs-tパスセットを求め,それらすべての組み合せを考慮する手法が代表的なものとして知られている[1]. Satyanarayana-Prabhakar は,s-tアサイクリックグラフ(文献[2]では p-acyclic subgraphs と表現)に着目し,正確な信頼度は各部分グラフの信頼度の代数和として計算できることを証明し,s-tアサイクリックグラフの生成アルゴリズム(以下では,パス除去法と記述)[2]を提案した。s-tアサイクリックグラフを取り上げる理由は,その数がs-tパスの組み合せの数に比して極端に少なくなることにある。しかしながら,彼らが提案したパス除去法は、

(i) グラフ G に含まれるサイクルが増える程その発生効率が低下する,

(ii) 初等的パス(ただ1つのs-tパスだけから成るs-tアサイクリックグラフ)の生成が最終レベルとなるため,任意のレベルでのs-tアサイクリックグラフの生成の打ち切りができず,信頼度の近似計算には不向きである。

という欠点を持っている。なお、s-tアサイクリックグラフの個数は、Gの規模の増大と共に指数的に増加するため、正確な信頼度を求めることは現実には不可能に近く、近似手法が有用となる。

本節では、上記の欠点を克服することを目的として構築した算法、深さ優先 探索 (DFS) を基本操作とするパス付加法、を提案する.

# 2.3.1 パス除去法[2]

Gにおいて、i個のs-tパスの和集合を $M_i$ とすると、Gのs-t信頼度R (G) は次式で与えられる。

$$R (G) = \sum_{i} \varepsilon_{i} P(M_{i})$$

$$\text{titl, } \varepsilon_{i} = (-1)^{i+1}, P(M_{i}) \triangleq \prod_{e \in M_{i}} P_{e} \text{Tibs}.$$

$$(2-11)$$

 $M_i$ はs-tグラフを構成するので、i番目のs-tアサイクリックグラフを $G_{a,i}=(V_{a,i},E_{a,i})$  、k番目のサイクリックグラフを $G_{c,k}=(V_{c,k},E_{c,k})$  とおけば、式(2-11)は、

$$R (G) = \sum_{i} d_{i} P (G_{a,i}) + \sum_{k} d_{k} P (G_{a,k})$$
 (2-12)

と書ける。ある与えられた $M_i$ ,  $M_j$  ( $i \neq j$ ) は同一のs = t グラフを構成することがあり、その関係によって $d_j$  ( $d_k$ ) が定められる。

Satyanarayana-Prabhakar および Willie は,式(2-12)のdi, d xが

$$d_{i} = (-1)^{|E_{a,i}| - |V_{a,i}| + 1}$$

$$d_{k} = 0$$
(2-13)\*

となることを証明している[2,3].

<sup>\*</sup>  $\mid E \mid$  は集合 Eの要素数を示す.

すなわち、式(2-11)は、次のように書ける.

$$R (G) = \sum_{i} d_{i} P (G_{a,i})$$
 (2-15)

式(2-15)は、Gが含むs-tアサイクリックグラフを列挙すれば、正確なs-t 信頼度が計算できることを示している。

パス除去法は、与えられたグラフGから出発し、順次、枝あるいは枝集合を開放除去しながら、重複なくs-tアサイクリックグラフを求めるように工夫されたものである。

アルゴリズムは次の3つの基本操作から構築されている。

- (1) Gがサイクルを含まなくなるまで、サイクルを構成する枝を1本ずっ開放除去し、アサイクリックグラフG を作る.
- (2) G'からs-tパス孤立枝をすべて開放除去し、さらに孤立頂点を取り除いて、s-tアサイクリックグラフ $G_i$ を作る。
- (3)  $G_i$ において、中立パスを1つ選び、 $G_i$ からそれに属する枝および中間頂点を除去する。得られたグラフを $G_i$ として、 $G_i$ がs-tパスとなるまでこの過程を繰り返す。

図2-4はパス除去法を ALGOL-like な言語で記述したものである。図2-4をグラフ G に適用するとき,得られる生成木の頂点は G , G' あるいは G , G は開放除去された G の枝あるいは枝集合に対応する。図2-4において,plevel は基本操作(2)の G , G を根(root)とする部分木の深さを示している。また,あるレベル(plevel)において,G は plevel 番目の頂点の兄(elder brother)あるいはその祖先(ancestor)の兄を終点とする木の枝の集合で,重複性のチェックに用いている。また,G は開放除去の候補となる枝あるいは枝集合である。

[例題2-1] 図2-1のグラフに対して、パス除去法を適用した結果(生成木)を図2-5に示す。 $\oplus$ 印の頂点がs-tアサイクリックグラフを、 $\bigcirc$ 印の頂点がそれ以外のグラフを示す。頂点  $G_{\mathfrak s}$ において、 $\mathfrak plevel=3$  、H=G-e-h-b 、 $A=\{a\}$  、 $B=\{(c,s)\}$  となっている\*。

<sup>\*</sup> G - e は Gから枝 e を開放除去することを意味する.

```
procedure S-T_ACYCLIC#1 (graph G, vertex s.t)
 begin
    procedure OPEN(subgraph H, arc set A, class of arc sets B, integer plevel)
     begin class of arc sets Q
            comment Q is local to procedure OPEN
         if plevel \neq 0 then
            begin
                output H;
                if H is not an s-t path then
                   begin
                      find neutral paths \{\mu_i\} in H;
                      Q := {\mu_i \mid \mu_i \cap A = \emptyset};
                           comment every neutral path of Q contains at least
                                      one path belonging to B provided that
                                      B + \phi and plevel > 1
                       while Q + \phi do
                          begin
                             select and delete any element \mu from Q ;
                             OPEN (H-\mu,A,Q,plevel+1);
                             A := A \cup \mu
                          end
                   end
            end:
         else
            begin
                if H contains a cycle \mu = \{e_1, e_2, \dots, e_i, \dots\} then
                   begin
                       Q := \{ e_i \in \mu \mid e_i \in A \}
while Q + \emptyset do
                          begin
                              select and delete any arc e from Q;
                              OPEN(H-e,A,Q,plevel);
                              A := A \cup \{e\}
                          end
                    end;
                else
                    begin
                       if H is not s-t acyclic then
                          begin
                              find s-t path isolated arcs y;
                              if y \cap A = \emptyset then OPEN (H-y, A, \emptyset, plevel+1)
                          end :
                       else OPEN(H,A, ø ,plevel+1)
                    end
             end
      end OPEN;
     OPEN (G, \phi, \phi, 0)
  end S-T_ACYCLIC#1 ;
```

図2-4 パス除去法

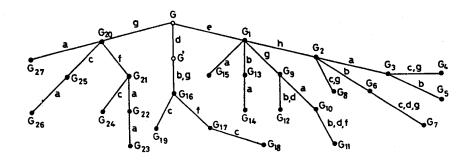


図2-5 例題 2-1の生成木

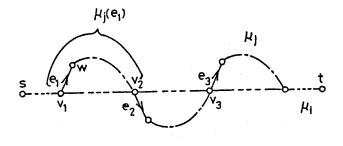


図2-6 付加パス

このアルゴリズムの時間複雑度,スペース複雑度について述べる.

計算時間は、Gがサイクルを含むか含まないかに依存することは容易に分かる。たとえば、互いに枝が素なサイクルがg個存在し、それらの長さがすべて等しく dis であると仮定すると、基本操作(1)が少なくとも $\sum_{r=0}^{\infty}dis'$ 回必要となる。s-t アサイクリックグラフの個数c が $c \ll \sum_{r=0}^{\infty}dis'$  の関係を持つならば、実行時間は dis' のオーダとなる。一方、 $c \gg \sum_{r=0}^{\infty}dis'$  あるいはGがサイクルを含まないグラフならば、基本操作(3)の実行時間に支配され、時間複雑度はO ( $c \cdot l_c$ ) となる。

スペース複雑度に関して最も支配的であるのはBであり,最初に作られたs - t アサイクリックグラフが含む中立パスの数は高々 $l_s$ 個であること,あるレベルでのs - t アサイクリックグラフの中立パスの数はその親が含む中立パスの数より少ないこと,に注意すればO( $l_s^2$ ) であることがわかる.

# 2.3.2 パス付加法

パス除去法に代わる新しいアルゴリズムの構築に当たって特に留意すべき点 は、次のことである。

- (i) 実行時間を1個当たり $O(n_c+l_c)$ とすること.
- (ii) 生成木の根から任意の距離 (レベル) でs-t アサイクリックグラフの発生を打ち切っても信頼度の上限値あるいは下限値が導けること.

# (1) s-tパスの発生と生成木

s-tパスはGの深さ優先探索によって求めることができる。いま, i番目に求まったs-tパスを $\mu_i$ とし, $\mu_i = \{\mu_1, \mu_2, \cdots, \mu_i\}$  とする。 $\mu_{i+1}$ は, $\mu_i$ (s, v) $=\mu_i$ (s, v)の関係を持つs-tパス $\mu_i$ ( $\ell$   $\mu_i$ )の内で, $\mu_i$ 上で t に最も近い頂点 vを持つものとして選ばれる。

s-tパス対 $\mu_i$ ,  $\mu_j$  ( $\in m_{i-1}$ ) について定義をしておこう.  $\mu_i$ ,  $\mu_j$ は一般にいくつかの頂点を共有する. いま、始点vが共有頂点で、終点wが $\mu_j$ だけに属する頂点であるような $\mu_j$ の枝e=(v,w)を対象とする. このような頂

点  $v \in \mu$   $_{i}$   $c \mu$   $_{i}$  の交差頂点と呼ぶ。交差頂点を s に近いものから  $v_{1}, v_{2}, \cdots$  と順序をつけるものとする。

 $e_1=(v_1,w)\in\mu$   $_j$ を $\mu$   $_i$ に関する $\mu$   $_j$ の付加枝と呼び, $\mu$   $_j$ の部分パス  $\mu$   $_j$   $_j$   $_j$   $_j$  を付加パスといい, $\mu$   $_j$   $_j$   $_j$  で表す(図2-6参照).  $\mu$   $_i$  に関する $\mu$   $_j$  の付加枝は, $\mu$   $_i$   $_j$  に属している他の s - t  $_j$  の付加枝ともなりうるので,以下では単に $\mu$   $_i$  に関する付加枝といい, $\mu$   $_i$   $_j$  で定まる $\mu$   $_i$  に関する付加枝の集合を $\theta$   $_i$  で表す.

Q,は有向木上で次のように説明できる: i番目の葉の兄あるいはその祖先の 兄を終点とする木の枝のラベルの集合である.

[例題 2-2] 図2-2において、 $Q_1 \sim Q_6$ はそれぞれ

$$Q_1 = \phi, \qquad Q_2 = \{f\},$$

$$Q_3 = \{a\}, Q_4 = \{a, f\},$$

$$Q_{5} = \{a, b\}, Q_{5} = \{a, b, h\}$$

となる. なお,  $Q_{5}$ ,  $Q_{6}$ における b は $\mu_{3}$ ,  $\mu_{4}$ に共通する付加枝である.

# - (2) 主定理とアルゴリズム

 $\{\mu\ (e)\ |\ e\in Q_i\}$  の部分集合に属する付加パスと $\mu_i$ によって構成されるs-tアサイクリックグラフの集合を $\beta(\mu_i,Q_i)$  で表す。

[補題2-1] 任意のi, j ( $i \neq j$ ) に対して,

$$\mathcal{G}(\mu_i, Q_i) \cap \mathcal{G}(\mu_j, Q_j) = \phi$$
 (2-16)  
が成立する.

(証明) いま、 $g_i \in \mathcal{G}(\mu_i, Q_i)$ ,  $g_j \in \mathcal{G}(\mu_j, Q_j)$  に対して、 $g_i = g_j$ と仮定しょう。また、一般性を失うことなくi < jとする。  $\mathcal{G}(\mu_j, Q_j)$  の作り方から明らかなように、 $g_i$ ,  $g_j$ のそれぞれは $\mu_i$ および $\mu_j$ を含んでいることになる。

 $\mu_1 \cup \mu_2$ が $s_1$ 側で作られる状況は図2-6のようになる(一般に交差頂点は少なくとも 1 個存在するが,個数が変っても一般性を失うことはない). $\mu_1$ に関して, $e_1$ , $e_2$ ,および $e_3$ は付加枝とならねばならない.しかし, $e_1$ が付加枝であることは, $\mu_1$ (s,  $\nu_1$ )と $e_1$ を含むすべてのs-tパスが $\mu_1$ が生成

される以前に調べ尽くされていることを意味する. すなわち,  $\mu \in \pi$ ; であることになり, i < jに反する. (証明終)

任意のs-tアサイクリックグラフは、1つのs-tパスといくつかの付加パスに分解できることから次の定理を得る。

[定理2-1] s-t アサイクリックグラフの集合 gは、

$$\mathcal{J} = \sum_{i=1}^{m} \mathcal{J} \left( \mu_{i}, Q_{i} \right) \tag{2-17} *$$

で表される.

なお、有向木から明らかなように、 $Q_1 = \phi$ であるから  $\mathcal{G}(\mu_1, Q_1) = \mu_1$  であることに注意しなければならない。

 $\mu$ の付加枝集合を $A_2$ ,付加枝集合の組み合せをとるために用いる集合を $A_1$ と定め, $A_1+A_2=\{e_1, e_2, \cdots, e_n\}$ とおけば,式(2-17)が次のように再帰的に記述できる.

$$\mathcal{J} = \mu_{1} + \mathcal{J}(\phi, \mu_{2}, \phi, Q_{2})$$
(2-18)  

$$\mathcal{J}(H, \mu, A_{1}, A_{2}) = \mathcal{J}(H, \mu', A_{1}, A_{2}') + (H + \mu)$$

$$+ \sum_{j=1}^{r} \mathcal{J}(H + \mu, \mu(e_{j}), \{e_{1}, e_{2}, \dots, e_{j-1}\}, \phi)$$
(2-19)

 $\mu$ はs-tパス $\mu$ ,かあるいは付加パス $\mu$ (e)であり, $\mu$ (e)の始点および終点はそれぞれs(e),最初に出会うH上の頂点である。また, $\mu$  は $\mu$  を深さ優先探索により修正したパスである。 $A_2$ ( $A_2$ )は $\mu$ ( $\mu$ )に関する付加枝の集合,すなわちsあるいはt(e)を根とする有向木の極大部分木に対して定まる付加枝の集合(ただし,Hに含まれる枝を除く)である。式(2-19)の第3項は $\{\mu$ (e) |e  $\in A_1+A_2\}$  からの組み合せをとるように工夫されている。

式(2-18)の第1項は $\mathcal{G}(\mu_1, \phi)$ ,第2項は $\Sigma_{i=2}^m \mathcal{G}(\mu_i, Q_i)$ に相当する、 $|Q_2|=1$ であることに注意して,式(2-18)の第2項を式(2-19)で展開す

<sup>\*</sup> Σは直和を意味する.

れば次式が得られる。ただし、 $Q_2 = \{e_2^1\}$  とする。

$$\mathcal{J}(\phi, \mu_{2}, \phi, Q_{2}) = \mathcal{J}(\phi, \mu_{3}, \phi, Q_{3}) + \mu_{2} + \mathcal{J}(\mu_{2}, \mu(e_{2}^{1}), \phi, \phi)$$
(2-20)

式(2-20)の第1項は $\Sigma_{i=3}$   $\mathcal{G}(\mu_i, Q_i)$ ,第2項以降は $\mathcal{G}(\mu_2, Q_2)$  に対応している。 $e_i$   $\in Q_i$   $(j=1, 2, 3, \cdots)$  とおき,式(2-20)の第1項に対し式(2-19)を繰返し適用すれば,式(2-18)が次のように展開できる。なお,i 行が  $\mathcal{G}(\mu_i, Q_i)$  に対応するように記述している。

$$\mathcal{J} = \mu_{1}$$

$$+ \mu_{2} + \mathcal{J}(\mu_{2}, \mu (e_{2}^{1}), \phi, \phi)$$

$$\vdots$$

$$+ \mu_{i} + \sum_{j} \mathcal{J}(\mu_{i}, \mu (e_{i}^{j}), \{e_{i}^{1}, e_{i}^{2}, \cdots, e_{i}^{j-1}\}, \phi)$$

$$\vdots$$

$$+ \mu_{m} + \sum_{j} \mathcal{J}(\mu_{m}, \mu (e_{m}^{j}), \{e_{m}^{1}, e_{m}^{2}, \cdots, e_{m}^{j-1}\}, \phi)$$

$$(2-21)$$

以上から,図2-7に示すような新しいアルゴリズム (パス付加法)を構築できる.

[例題2-3] 図2-1のグラフに対して、パス付加法を適用した結果(生成木)を図2-8に示す。生成木の根はs、tのみから成るグラフに、それ以外の頂点はs-tアサイクリックグラフ(s-tパス+ $\Sigma$ 付加パス)に対応している。ここで、パス付加法の時間複雑度、スペース複雑度について述べる。

```
procedure S-T_ACYCLIC=2 (graph G, vertex s,t)
 begin
     procedure ADD(subgraph H, path \mu, arc set A_1, A_2, integer i)
      begin arc set A.A3
                 comment A and A3 are local to procedure ADD
          output H+\mu ;
         A := A_1 + A_2;

A_3 := \emptyset;
         whlie A + Ø do
             begin
                 select and delete any arc e from A:
                 if there is a path \mu(e) which contains
                        no cyclic arc with respect to H then
                     begin
                        ADD(H+\mu,\mu(e),A<sub>3</sub>,\phi,i+1);
                        A_3 := A_3 + \{e\}
                     end
             end;
         if a new path \mu^{\boldsymbol{\cdot}} is found in the course of
                                backtracking on \mu then
             begin
                 modify A_2 to A_2; ADD (H, \mu^*, A_1, A_2^*, i)
             end
     end ADD;
    find an s-t path \mu :
 ADD(\phi ,\mu,\phi ,\phi ,1)
end S-T_ACYCLIC=2;
```

図2-7 パス付加法

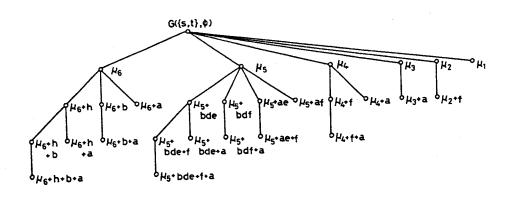


図2-8 例題 2-4の生成木

生成しようとする場合であり、いずれもサイクル $\{4, 8, 2, d, 3, e, 4\}$ が $\mu$  に含まれることになるからである。なお、この例では $\mu$ (e)の生成が不成功に終わるとなることはない。

記憶領域を支配するのは $A_1$ および $A_2$ であるが,これらはパス除去法のBと同様 O ( $l_c^2$ ) である.

# 2. 4 木展開法

本節では、得られる積項数がSDP法のうち最も少ないと考えられる Abraham の算法[15]のそれと同程度であり、かつ演算の高速化を目的としたグラフ理論的算法(木展開法)を提案する.

# 2.4.1 基準木と積項木

2. 3節において定義した付加枝はs-tパスを生成する過程 (DFS) において記憶しておくことが可能であり、付加枝がわかれば付加パスの生成も容易である.

図2-2の有向木 Tに関して、s-tパス $\mu_5$ の付加パス $\mu_{15}=af$ 、 $\mu_{25}=ae$ 、 $\mu_{35}=bdf$ 、 $\mu_{45}=bde$  は、その生成順に従い図2-9のように表せる。これを5番目の基準木といい、 $T_5$ で表す。同様に、 $T_i$  (i=2, 3, 4, 6)を描けば図2-10を得る。

$$T_{5} = X_{5} (X_{f} + X_{e}) + X_{5} X_{d} (X_{f} + X_{e})$$
 (2-22)

と表せば、 $F_{5}$ は式(2-7)を参照して次のように書くことができる。

$$F_5 = \overline{T}_5 A_5 \tag{2-23}$$

このことから、一般に $F_R$ に関して次の定理が成立する。

[定理2-2]  $F_R$ は次式で表すことができる.

$$F_{R} = A_{1} + T_{2}A_{2} + T_{3}A_{3} + \cdots + T_{m}A_{m}$$
 (2-24)

式(2-9)を用いて、 $T_{\bullet}$ を展開すれば次式が得られる.

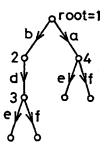


図2-9 基準木 / 5

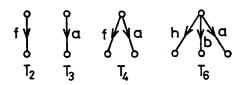


図2-10 基準木

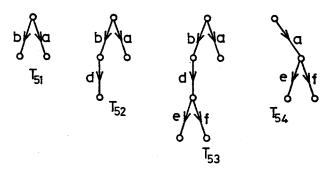


図2-11 / 7.の部分木

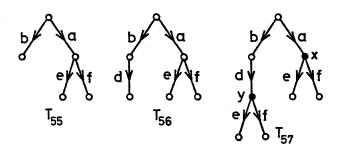


図2-12 / 7.の他の部分木

 $\overline{T}_{5} = \overline{X_{a} (X_{f} + X_{e}) + X_{b} X_{d} (X_{f} + X_{e})}$ 

- $= (\bar{X}_b + X_b \bar{X}_f \bar{X}_b) (\bar{X}_b + X_b \bar{X}_d + X_b X_d \bar{X}_f \bar{X}_b)$
- $= \bar{X}_a \, \bar{X}_b + \bar{X}_a \, X_b \, \bar{X}_d + \bar{X}_a \, X_b \, X_d \, \bar{X}_f \, \bar{X}_e$

$$+ X_b \overline{X}_f \overline{X}_e \left( \overline{X}_b + X_b \overline{X}_d + X_b X_d \overline{X}_f \overline{X}_e \right)$$
 (2-25)

 $= \bar{X}_a \, \bar{X}_b + \bar{X}_a \, X_b \, \bar{X}_d + \bar{X}_a \, X_b \, X_d \, \bar{X}_f \, \bar{X}_e$ 

$$+ x_b \overline{X}_f \overline{X}_e \left( \overline{X}_b + X_b \overline{X}_d + X_b X_d \right) \tag{2-26}$$

 $= \bar{X}_a \, \bar{X}_b + \bar{X}_a \, X_b \, \bar{X}_d + \bar{X}_a \, X_b \, X_d \, \bar{X}_f \, \bar{X}_e + X_a \, \bar{X}_f \, \bar{X}_e \, (\bar{X}_b + X_b)$ 

(2-27)

$$= \bar{X}_{s} \bar{X}_{b} + \bar{X}_{s} X_{b} \bar{X}_{d} + \bar{X}_{s} X_{b} X_{d} \bar{X}_{f} \bar{X}_{e} + X_{s} \bar{X}_{f} \bar{X}_{e}$$
 (2-28)

式(2-28)のi (i=1, 2, 3, 4) 項は図2-11のように $T_{\mathfrak{s}}$ の部分木 $T_{\mathfrak{s},i}$  (i=1, 2, 3, 4) として表すことができる.ここで,各積項中において,補元となる変数は対応する部分木で葉に接続する枝である.また,式(2-25)の第4項を展開して得られる3個の積項もそれぞれ $T_{\mathfrak{s}}$ の部分木 $T_{\mathfrak{s},i}$  (i=5, 6, 7) に対応させることができる(図2-12参照).

図2-11および図2-12の枝および頂点に対して,元のグラフ(図2-1)のラベルを付ければ,図2-11の各部分木の枝のラベルはすべて異なるが,図2-12の $T_{57}$ には同一のラベルを持つ枝が存在する.この同一ラベルを持つ枝の始点は元のグラフにおいて同一の頂点である.この頂点をx, yと名付ける( $\oplus$ 印の頂点).式(2-25)から式(2-26)の展開において,共通因子 $\overline{x_1}$ ,  $\overline{x_2}$ , が除去されている.これは, $T_{57}$ のyに接続される枝fおよびeとその終点(葉)を開放除去することに対応しており,その結果 $T_{55}$ が得られる.さらに,式(2-26)から(2-27)への展開は,葉yを終点とする枝f0開放除去により $f_{55}$ から $f_{55}$ を求めることに対応している.同様にして,式(2-27) $f_{55}$ 0円 になっている.

このように式の展開に対応して $T_1$ の枝および終点を開放除去することを木の縮約という。また、式(2-28)の各項に対応する $T_8$ の部分木を積項木と名づける。

[定義2-4]  $T_i$ において,頂点 $\nu$ の縮約は $T_i$ に対して以下の手続きを行うことと同等である.

(i) y を根に持つ  $T_i$ の極大部分木  $T_{iy}$ を除去する.

- (ii) νを終点とする直列枝およびνを除去する.
- (iii) (ii) によってある頂点が葉となるとき、その頂点をyとおいて(ii) を繰返す。

 $T_i$ に関する部分木(積項木)は次のような性質を持っている.

- (i) 積項木は $T_i$ の根を含み連結である.
- (ii) 積項木を構成するどの頂点も、縮約された T ,上における兄弟をすべて含む。

## 2.4.2 基準木の縮約のグラフ理論的条件

T,の頂点および枝は、グラフG上の頂点および枝に対応させることができるので、それぞれGの頂点、枝に相当するラベルを付けるものとする.

[定理2-3] I,上のある頂点x, yにおいて,その子の接続される木の枝のラベルの集合 $E_x$ ,  $E_y$ が $E_x \supseteq E_y$ を満たすとき,頂点yの縮約が可能である.

(証明) 図2-13はx, yを含むT, rの部分木で,根からx。は共通の祖先,r0, r1, r2, r3, r3, r3, r3, r4, r5, r5, r6, r7, r7, r8, r9, r9,

 $\cdots X_{a1} \cdots X_{a2} \bar{E}_X$ 

を含む項がある。ただし, $\bar{E}_x = \bar{X}_{e1} \bar{X}_{e2} \cdots \bar{X}_{er}$ , $e_z \in E_x$ とする.一方, $T_{ib}$ に関する表現には,

...  $X_{b1}$  ...  $\bar{X}_{b4}$   $(\bar{X}_{b2} + X_{b2} \bar{X}_{b3} + X_{b2} X_{b3} \bar{E}_{\nu})$ 

を含む項がある。したがって、 $T_{i,s} \cup T_{i,s}$ に関するブール表現には、両項の積

 $\cdots$   $(X_{a1}$   $\cdots$   $X_{a2}\bar{E}_x)$   $(X_{b1}$   $\cdots$   $\bar{X}_{b4}$   $(\bar{X}_{b2} + X_{b2}\bar{X}_{b3} + X_{b2}X_{b3}\bar{E}_y)$ )を含むものがある。 $E_x \supseteq E_y$ であれば,上式は,

 $\cdots (X_{s1} \cdots X_{s2} \bar{E}_x) (X_{b1} \cdots \bar{X}_{b4} (\bar{X}_{b2} + X_{b2} \bar{X}_{b3} + X_{b2} X_{b3}))$   $= \cdots (X_{s1} \cdots X_{s2} \bar{E}_x) (X_{b1} \cdots \bar{X}_{b4})$ 

となり、2を根とする左部分木が2の縮約によって欠落したものと等しくなる.

(証明終)

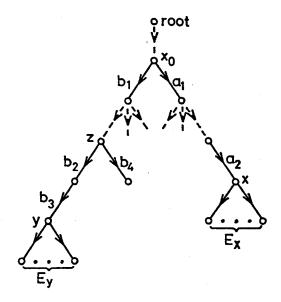


図2-13 定理2-3の説明図

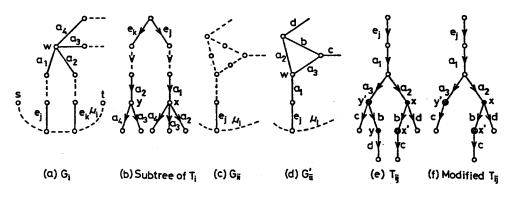


図2-14 定理2-4の説明図

 $E_x \supseteq E_y$ の関係を持つ頂点yはxと同じラベルを持つ頂点であることは明かである。そこで、指定したある頂点xに対して、根からxに至るパス上の頂点の弟とその子孫の内で、xと同じラベルを持つものの集合をL(x)とする。

いま, $\mu$  ,上の頂点をsからtに向かって順番を付け,この番号をN(・)で表す.また,付加枝 $e \in Q$  ,の始点,終点をそれぞれs(e),t(e)で表し,Q ,に属する付加枝を次のように分類する.ただし, $r = |Q_1|$ とする.

 $N(s(e_1)) \leq N(s(e_2)) \leq \cdots \leq N(s(e_r))$ 

[定理2-4] 根に対して、ラベルt ( $e_1$ )、t ( $e_2$ )、…、t ( $e_n$ )の頂点をそれぞれ1、2、…、r と順序を付けた  $T_n$ において、すべての  $y \in L$  (x) に対して、 $E_x \supseteq E_y$ が成立する。

(証明) Gは無向グラフとし、ラベルt(e)の頂点を根とするT1の極大部分木をT1.7で表す。

(i)  $x \in T_{i,j}, y \in T_{i,k} (j < k)$  の場合

2 つの付加枝  $e_{i}$ ,  $e_{k}$ で定まる付加パスが頂点を共有する場合の例を図2-14 (a)に示す. ここで、共有する頂点のうちsに最も近いものをwとする.

これより  $T_i$ の関連部分を示せば図2-14(b)となる。  $E_v$ が  $a_1$ を含まないのは, $\mu_i$ の一部と( $e_k$ , …,  $a_2$ ),( $a_1$ , …,  $e_i$ )がサイクルを構成するからである.

したがって, $E_x \supseteq E_y$ が成立する.なお,等号はs  $(e_j) = s$   $(e_k)$  のとき成立する.

(ii)  $x, y \in T_{i,j}$ の場合

e を付加枝とする付加パスが頂点を共有する一般的な状況を図2-14(c)に示す。 さらに、この図は図2-14(d)のように変換しても以下の議論の一般性を失うことはない。

また, $T_i$ の関連部分を図2-14(e)に示す. $\blacksquare$ 印の頂点をx,yにとれば, $E_x$ つ  $E_y$ の関係が成立している.

一方, $\bigcirc$ 印の頂点をx′,y′にとれば, $E_{x'}$   $\subset E_{y'}$  の関係が成立し, $E_{x'}$   $\not$   $\not$   $\not$   $\not$   $\not$   $\not$   $\not$   $\not$  v となっている。しかし,x の頂点を根に近いものから優先してとることにすれば,この矛盾は図2-14(f)に示すように解消される。x の選択によりy の

縮約が起こり、y'を根とする右部分木が欠落し、その結果x', y'の間に  $E_{x'} = E_{y'}$ の関係を生じる.

また、 xを y, yを xと交換した状況においても同様の結果が得られる.

(証明終)

定理 2-4 の後半は,付加枝の順序が保存される限り,  $T_i$  の生成過程(DFS)における枝の選択には何の拘束もないことを意味している.

以上の議論から、指定した頂点xに関する木の縮約は、すべての頂点 $y \in L(x)$ に対して行えばよいことが分かる。

# 2.4.3 積項木の生成

葉以外の頂点の集合をX, L (x) ( $x \in X$ ) の縮約によって基準木 T, から作られる木を T, (X) と表せば、積項木は次のように生成される

- (1)  $T_i$ の根とその子だけから作られる積項木を  $T_{i+1}$ とし,  $T_i$  ( $X_1$  = {root}) :=  $T_i$ とおく.
- (2)  $T_i(X_j)$ ,  $T_{i,j}$ から積項木 $T_{i,j+1}$ の生成
  - (i)  $T_{i,j}$ 上において、左優先で葉xを選ぶ、
  - (ii)  $T_i(X_i)$  上で x の子が存在する場合
    - (a)  $y \in L$  (x) について  $T_i$  ( $X_j$ ) を縮約し、  $T_i$  ( $X_{j+1}$ )  $riangleq T_i$  ( $X_j \cup \{x\}$ ) を作る.
    - (b)  $T_{i,j}$ 上のxに対して, $T_{i}$ ( $X_{j}$ )上にあるxの子を付け,かつ $X_{j}$ の各頂点が $T_{i}$ ( $X_{j+1}$ )に関して積項木の性質(ii)を満たすように $T_{i,j}$ を修正し, $T_{i,j+1}$ を作る。j:=j+1とおいて(i)へ.
  - (iii)  $T_i(X_i)$  上でxの子が存在しない場合
    - (a)  $T_{i,j}$ からxおよびxを終点とする枝を除去したものを $T_{i,j}$ とおき、xの親をvとおく.
    - (b)  $v \in x$ 以外の子があるば(i)へ.
    - (c) vに x以外の子がない場合

# vが根ならば操作は終了し、vが根でなければ、 $X_J$ : = $X_J$ - $\{v\}$ 、x: = vとおいて (ii) - (a) へ.

図2-9の  $T_{5}$ から積項木が生成される状況を表2-2に示す。  $T_{i,j}$ の枝集合として格納される枝の順序は,  $T_{i}$ をDFSにより右側優先で探索したときの枝の探索順序に矛盾しないものとする。このようにすると,少ない処理で次の積項木を生成することができる。また,  $T_{i}$  ( X ) は  $T_{i}$ 上で縮約した頂点にマークを

表2-2 / 6の積項木

j	$X_{J}$	$T_{i}(X_{j})$	T ; , ;の枝集合	積項木	積項
1	{1}	Т 5	a, b	T 5 1	$\bar{X}_a  \bar{X}_b$
2	{1, 2}	Т 5	a, b, d	T 5 2	$\overline{X}_b X_b \overline{X}_d$
3	{1, 2, 3}	Т 5	a, b, d, f, e	T 5 3	X. X b X d X f X
	{1, 2, 3}	Т 5	a, b, d, f		
	{1, 2}	Т 5	a, b, d		
	{1}	Т 5	a, b		
	{1}	Т в	а		
4	{1, 4}	T 5 4	a, f, e	T 5 4	$X_{\theta} \overline{X}_{f} \overline{X}_{\theta}$

付与するだけで表すことができる.

 $T_i$  (i=2, 3, 4, 6) はそれ自身で積項木となるため、 $F_R$ は次のように導ける。

$$F_{R} = A_{1} + \overline{X}_{f} A_{2} + \overline{X}_{o} A_{3} + \overline{X}_{o} \overline{X}_{f} A_{4}$$

$$+ (\overline{X}_{o} \overline{X}_{b} + \overline{X}_{o} X_{b} \overline{X}_{d} + \overline{X}_{o} X_{b} X_{d} \overline{X}_{f} \overline{X}_{e} + X_{o} \overline{X}_{f} \overline{X}_{e}) A_{5}$$

$$+ \overline{X}_{o} \overline{X}_{b} \overline{X}_{b} A_{5} \qquad (2-29)$$

## 2.5 パスの処理順序と積項数

表現式の積項数は,TR法[2,4,11]のように枝の探索順序には全く影響を受けない算法もあるが,一般的には枝あるいはパスの処理順序に大きな影響を受ける。本節では,パスの処理順序と積項数の関係を調べ,その結果を用いて改良木展開法を提案する.

図2-1のグラフに対して,

- (i) SDP法(枝数の少ないパスから優先的に演算を行っている)[15],
- (ii) 生成順に従いパスを処理するように変更したSDP法,
- (iii) 枝数の多いパスから優先的に演算を行うように変更したSDP法を用いて表現式を導けば、その積項数はそれぞれ(i)が8,(ii)が11,(iii)が29であった。ここで、(i)をB-SDP法(Best selecting-SDP method)、(ii)をR-SDP法(Random numbering-SDP method)、(iii)をW-SDP法(Worst selecting-SDP method)と呼ぶことにする。

木展開法におけるパス $\mu_i$ に関する基準木 $T_i$ の生成には,DFSにより既に生成されているパス $\mu_i$ (i=1, 2, …, i-1)の情報を必要とするため,生成順と無関係な順序でパスを処理することはできない.したがって,本節ではDFSによるパスの生成という制約の上で,可能な限り枝数の少ない(あるいは多い)パスを優先的に算出することを試みる.

頂点iからtに至るすべてのパスのうち,枝数が最も少ないパスを,頂点iからtに至る最短パスという。また,最短パスの枝数を,頂点iからtへの距離,あるいは頂点iの重みと呼ぶ。なお,頂点tの重みを0と定める。

DFSによりパスを成長させる過程において、小さな(あるいは大きな)重みを持つ頂点を終点として持つ枝を優先的に選択すれば、2.4節で生成されているパスとはその生成順序が異なり、かつこれらパスに木展開法を適用して得られる表現式の積項数は、2.4節に比べて減少(あるいは増加)することが予想できる。

このことを図2-1を用いて示そう.図2-1の各頂点に対して重みを付加したグラフを図2-15に示す(頂点の重みをそれぞれカッコ内の数値で示す).

小さな重みを持つ頂点を終点とする枝を優先的に取り上げて生成されたs-tパスを,その生成順にしたがい右側優先の有向木Tで表せば図2-16を得る。図2-16のパスのそれぞれに対して生成される基準木を図2-17に示す。図2-17の $T'_2$ , $T'_4$ および $T'_5$ は各々それ自身積項木であるので,それ以外の基準木 $T'_3$ および $T'_5$ の積項木を図2-18に示す。図2-17および図2-18から信頼度表現式 $F'_8$ が次のように導かれる。

$$F'_{R} = A'_{1} + \overline{X}_{f} A'_{2} + (\overline{X}_{a} + X_{a} \overline{X}_{e} \overline{X}_{f}) A'_{3} + \overline{X}_{a} \overline{X}_{h} A'_{4}$$

$$+ (\overline{X}_{a} \overline{X}_{c} + \overline{X}_{a} X_{c} \overline{X}_{g} \overline{X}_{h}) A'_{5} + \overline{X}_{a} \overline{X}_{c} \overline{X}_{f} A'_{6}$$

$$|F'_{R}| = 8$$
(2-30)

これをB-木展開法と呼ぶことにし、2.4節の木展開法をR-木展開法と呼ぶことにする。

一方、大きな重みを持つ頂点を優先的に選択して得られたs-tパスを、その生成順に従い図2-19に、パスに関する基準木を図2-20に示す。さらに、図2-20のT"。を除く基準木のそれぞれの積項木を図2-21に示す。図2-20および図2-21からF"。が次のように導かれる。

$$F_{R} = A_{1} + (\bar{X}_{e} + X_{e} \bar{X}_{h}) A_{2} + (\bar{X}_{b} + X_{b} \bar{X}_{d}) A_{3}$$

$$+ (\bar{X}_{e} \bar{X}_{b} + X_{e} \bar{X}_{h} \bar{X}_{b} + \bar{X}_{e} X_{b} \bar{X}_{d} + X_{e} \bar{X}_{h} X_{b} \bar{X}_{d}) A_{4}^{"}$$

$$+ \bar{X}_{a} \bar{X}_{b} A_{5}^{"} + (\bar{X}_{g} \bar{X}_{a} \bar{X}_{b} + X_{g} \bar{X}_{d} \bar{X}_{a} \bar{X}_{b} + X_{g} X_{d} \bar{X}_{f} \bar{X}_{a} \bar{X}_{b}$$

$$+ X_{a} \bar{X}_{e} \bar{X}_{f} \bar{X}_{b} + \bar{X}_{a} X_{b} \bar{X}_{d} + X_{a} \bar{X}_{e} \bar{X}_{f} X_{b} \bar{X}_{d}$$

$$+ X_{b} X_{d} \bar{X}_{e} \bar{X}_{f}) A_{6}^{"} \qquad (2-32)$$

$$|F_{R}^{"}| = 1.7$$

このような展開法をW-木展開法と呼ぶことにする.

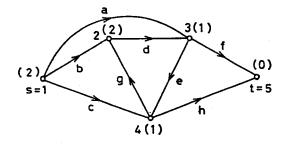


図2-15 重みを付加したグラフ

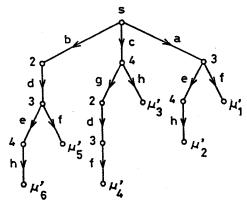


図2-16 図2-15の有向木 T'

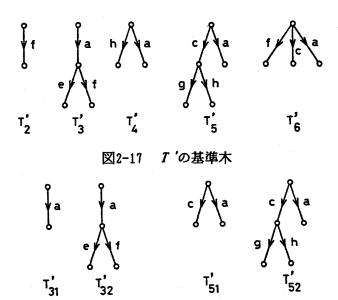
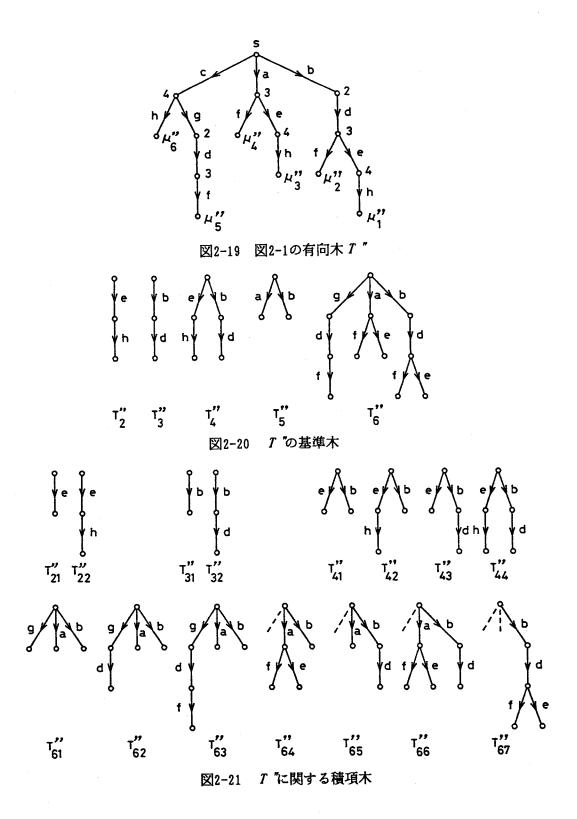


図2-18 T'sとT'sの積項木



パスの生成順序と積項数の関係を調べるため,図2-22のグラフ(文献[2]の Fig. 8 のグラフであり,かつ2.6 節実験結果と考察で述べる番号2のグラフである)のsとtを除く頂点にランダムな値を付与し,小さな値を持つ頂点から順に2から8までの番号を頂点番号として割り当てることにする。なお,頂点の重みをそれぞれカッコ内の数値で示す。この割り当てを行ったグラフに対して,R-木展開法を適用し積項数を算出した。1 回の割り当てを1 回のテストとし,合計 1000 回のテストを行った。このとき得られた積項数の分布を表2-3に示す。また,図2-22のグラフに対して,上述の3種類のSDP法および木展開法を適用して得られた結果を表2-4に示す。

表2-3および表2-4から次のことがいえる.

- (i) SDP法および木展開法において、パスの処理順序が変われば、得られる積項数は大きく変動する.
- (ii) 頂点番号の付与の仕方によって、R-木展開法の積項数は 250 ~ 468 と変動する.したがって、表2-4および2.4節における積項数 257 が常に得られるとは限らない.
- (iii) 積項数の削減を目的として設計されたB-木展開法は、表2-3の下限値に近い値の積項数を算出しているので、本節の目的を達成している。そこで、このB-木展開法を改良木展開法と呼ぶことにする。

# 2.6 実験結果と考察

計算機実験に用いたテストグラフは12個のサイクルを含むグラフ(下記のタイプ $I \sim V$ ), と3個のアサイクリックグラフ(下記のタイプVI)である.

#### (i) タイプ I

文献[2]で取り上げている2つのグラフである。番号1のグラフはARPAネットワークのモデルである。また、番号1および2のグラフは互いに枝が素なサイクルをそれぞれ8個、2個含んでいる。

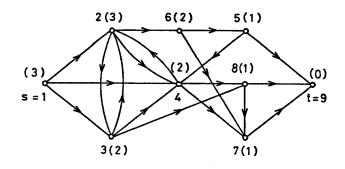


図2-22 Satyanarayana のグラフ

表2-3 図2-22の積項数

積	項	数	頻度
250	~	279	74
280	~	299	94
300	~	319	157
320	~	339	170
340	~	359	185
360	~	379	74
380	~	399	44
400	~	419	102
420	~	439	63
440	~	468	37

表2-4 積項数の分布

	DP法			木展開法		
	w	R	В	w	R	В
積項数	1247	263	236	458	257	257

表2-5 DSPのサイクル数

互いに枝が素なサイクル数
6
6
8

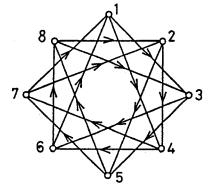


図2-23 DSP G(8, {2,3})

#### (ii) タイプⅡ

表2-5に示すDSP[5,6]を次の(a),(b) のように変形したグラフである。 参考のため, $n_G=8$ ,シンボル $\{2,3\}$ のDSP $G(8,\{2,3\})$ を 図2-23に示す。

- (a) DSPの任意の頂点をs, tとする.
- (b)  $d^-(s) = d^+(t) = 0$  となるようにDSPの枝を開放除去する. なお、DSPは最適通信網の1つのクラスであることを付記しておく.
  - (iii) タイプⅢ

表2-5のDSPを次の(a), (b)のように変形したグラフである.

- (a) DSPの頂点以外にs, tを付加する.
- (b)  $d^+(s) = d^-(t) = 1$ となるように枝を付加する.
- (iv) タイプⅣ

文献[7]を用いて発生した有向グラフを次のように変形して得られるグラフである.

- (a) 有向グラフの頂点以外にs, tを付加する.
- (b) sからグラフ内のソース頂点,またグラフ内のターミナル頂点から tに至る有向枝を付加する.
- (v) タイプV

文献[7]を用いて発生した無向グラフを次のように変形して得られるグラフである。

- (a) 任意の頂点をs. tとする.
- (b) 無向枝を互いに逆向きの2本の有向枝に置き換える.
- (c)  $d^{-}(s) = d^{+}(t) = 0$ となるように枝を開放除去する.
- (vi) タイプVI

番号13のグラフは梯子状グラフ,14は格子状グラフ,15は14の各格子に対角枝を1本ずつ付加したグラフである.

なお,計算機実験は九州大学大型計算機センター FACOM M-382 を使用し,プログラム言語として FORTRAN を用いた.

# 2.6.1 パス除去法とパス付加法の比較

番号1~15のグラフを対象として、パス除去法[2](#1で示す)およびパス付加法(#2)に関する比較実験を行い、表2-6の結果が得られた。なお、番号13~15のグラフはアサイクリックグラフであるため、パス除去法では基本操作(1)および(2)を、またパス付加法ではサイクルチェックのための祖先を作る作業を省略して実行した。表2-6より次のことがいえる。

番号1,10のグラフおよびタイプ $\Pi$ ,Vのグラフにおいて、パス付加法の実行時間がパス除去法のそれに比べ著しく低減されている。その理由は次のように考えられる。

G(9, {1,3,4})を例に用いて述べる。このグラフが含む互いに枝が素なサイクルの数は、長さ3のものが7個、長さ6のものが1個の計8個である。したがって、2.3.1節で述べた基本操作(1)の処理の回数は、少なくとも次式で与えられる。

$$6 \cdot \sum_{i=0}^{7} 3^{i} + 1 = 1 \ 9 \ 6 \ 8 \ 1$$

この数はs-t アサイクリックグラフの個数に比して大きい。したがって、パス除去法は実行時間の多くがサイクル除去に費やされるため、パス付加法に比べ多くの演算時間を必要とする。

#### 2.6.2 木展開法とSDP法の比較

パス付加によるTR法(#2で示す), SDP法(Aggarwal の算法[14](#3), Abraham の算法[15](#4)), および木展開法(#5)に対する比較実験の結果を表2-7に示す。

この表から次のことがいえる.

- (i) #4 および #5 は他の算法よりも得られる積項数が少ない.
- (ii) #5が他のどの算法よりも高速である。特に、s-tパス数が多い グラフ(番号4,5,9,13,15のグラフ)において顕著である。

表2-6 パス除去法とパス付加法の比較

タイプ	グラフ	グラフ	s – t アサイク	CPU時間	(msec)
	番号	の規模	リックグラフ数		
		Па, Са, 🛮		#1	#2
I	1	8, 20, 24	495	81	27
	2	9, 19, 35	5287	137	203
I	3	7, 16, 25	1143	36	51
	4	8, 19, 58	22231	531	814
	5	9, 22, 68	20231	634	813
Ш	6	9, 23, 27	1543	847	82
	7	10, 26, 34	2711	3329	157
	8	11, 29, 19	919	15557	81
IV	9	17, 32, 80	362751	8988	16863
	10	17, 32, 10	351	6227	25
v	11	15, 36, 26	1087	37906	94
	12	15, 35, 8	191	21802	17
VI	13	14, 23, 70	14679	413	411
	14	16, 24, 20	2895	84	130
	15	16, 33, 63	3327103	78521	110215

表2-7 木展開法とSDP法の比較

グラフ	積項数			CPU時間(msec)				
番号	#2	#3	#4	#5	#2	#3	#4	#5
1	495	113	67	62	27	29	7	3
2	5287	1272	236	257	203	449	27	8
3	1143	318	54	78	51	102	6	3
4	22231	3828	358	369	814	2557	87	14
5	20231	6404	353	558	813	4919	77	20
6	1543	269	63	83	82	106	9	5
7	2711	681	84	126	157	390	16	6
8	919	1882	27	76	81	792	6	5
9	362751	_	6136	13400	16863	-	2372	321
10	351	59	27	426	25	22	3	12
11	1087	1133	205	369	94	382	20	10
12	191	120	65	149	17	22	2	3
13	14679	36939	1513	1050	411	23241	547	28
14	2895	28518	1435	1435	130	11422	107	35
15	3327103	-	15032	13104	110215	_	2981	452

<sup>-</sup> は積項数が多すぎるため、演算を途中で打ち切ったことを示す。

# 2.6.3 パスの処理順序と積項数の関係

SDP法[15](#4), 木展開法(#5), および改良木展開法(#6)に対して行った実験結果を表2-8に示す.

その結果、表2-8から次のことがいえる。

- (i) #5および#6において,得られる積項数がほぼ同一であるとき, 演算時間もほぼ同一である。したがって,DFSの途中における枝の 選択に必要な時間が積項生成に必要な時間に比べ非常に小さく,#6 の高速性は#5と同様保存されている。
- (ii) 番号5,9,および10のグラフのように,#5の積項数が#4に 比べて多い場合,#6において積項数の大幅な削減が行われる.この とき,積項数の削減にともなって,#6は実行時間を大幅に減少でき, いずれの算法よりも高速なものとなる.

# 2.7 結言

本章では、ネットワークのs-t信頼度を求める3つの算法を提案した。

(1) 深さ優先探索によるs-tパスの発生を基本操作とし、これらのパスの組み合せによる重複のないs-tアサイクリックグラフの生成アルゴリズムを提案した。

このパス付加法は次のような特徴を持っている.

- (i) パス除去法では,実行時間が,互いに枝が素なサイクルの数によって大きく左右される場合があるが,このアルゴリズムではほとんどその影響を受けない.しかも,そのようなサイクルの数が少ない場合でも両者の実行時間は実験の範囲内では大差がなかった.
- (ii) 式(2-11)の $M_i$ と生成木のレベルiでのs-tサイクリックグラフを対応させれば信頼度表現が導かれる。このことより,奇(偶)数レベルまでのすべてのアサイクリックグラフを用いて信頼度の上限(下限)値が計算できる。

表2-8 パスの処理順序と積項数の関係

	1			1			
グラフ	積項数			CPU時間(msec)			
番号	#4	#5	l #6	#4	#5	#6	
				ļ			
1	67	62	62	7	3	3	
2	236	257	257	27	8	8	
3	54	78	69	6	3	3	
4	358	369	346	87	14	15	
5	353	558	351	77	20	14	
6	63	83	75	9	5	5	
7	84	126	126	16	6	8	
8	27	76	62	6	5	7	
9	6136	13400	3863	2372	321	101	
10	27	426	42	3	12	3	
11	205	369	189	20	10	9	
12	65	149	72	2	3	2	
13	1513	1050	828	547	28	23	
14	1435	1435	1435	107	35	35	
15	15032	13104	12468	2981	452	438	
		<u> </u>	· ·				

- (iii)  $g(\mu_1, Q_1)$ , …,  $g(\mu_i, Q_i)$  ( $i \leq q$ ) に属するすべての s t アサイクリックグラフで導かれる信頼度表現は,繰返し I E 法, S D P 法におけるステップ i での信頼度表現と等価であることが明か にされている[10]. このことより,ステップ i での信頼度表現から 1 つの下限値が計算できる.
- (2) s-tパスの集合から信頼度表現式を導出することに関して、SDP法は $\overline{A_1}$   $\overline{A_2}$  …  $\overline{A_{i-1}}$   $A_i$  を互いに素な積項の和に展開する算法であるため、展開の途中で生じる積項をすべて記憶する必要があること、また演算時間が展開のための比較回数によって大きく支配される等の欠点を持つことを示した。 さらに、 $\mu_i$ 上に始点および終点を持つ $\mu_i$  (i=1, 2, …, i-1) の部分パスから1つの根付木(基準木)を生成するとき、その部分木(積項木)から信頼度表現式の1つの積項を直接導くことができる木展開法を提案した.

また、パス付加法、SDP法との比較実験を行い、木展開法が積項数ではSDP法と同程度であり、実行時間では最良の算法であることを示した。

(3) 信頼度表現式の積項数は、TR法のように頂点および枝の探索順序によって全く影響されない算法もあるが、一般にはその処理順序(あるいはパスの処理順序)によって大きく変動する。与えられたグラフ上の各頂点から t に至る最短パスを求め、この枝数をその頂点の重みと定めるとき、小さな重みを持つ頂点を終点とする枝を優先的に選択しながらパスを生成し、その生成順序に従って木展開法を適用すれば、積項数の削減が可能であることを示した。

また、SDP法および木展開法に関して種々の実験を行い、次の結論が得られた。

- (i) 2.5節で提案する改良木展開法は得られる積項数が最も少ない.
- (ii) 改良木展開法は他のどの算法よりも高速である.

以上より、提案した3つの算法および既存の算法のうち、改良木展開法が最 も優れた算法であることが判明した.

# 第3章 ネットワークのソース・全端子信頼度

#### 3.1 緒言

通信網,電力網,交通網などのネットワークのように,ある既知の故障確率 (非信頼確率)のもとで動作する要素を構成成分とするシステムが与えられて いるとき,それらの要素を頂点,枝に対応させて,システム全体を信頼性グラフ(あるいは確率グラフ)として表現することができる[1].

信頼度問題の最も基本的な問題は、グラフ上の指定した 2 頂点 s, t に対して、s から t への到達可能性を問題とする s - t 信頼度問題であり、現在までに多くの研究がなされている [2,13,14,15].

一方,近年のコンピュータネットワークなどの情報通信ネットワークの発達は,s-t 信頼度問題よりももっと総体的な信頼度の計算を必要としている.この種の問題は,グラフ上の指定した頂点 s からそれ以外の複数個の頂点の集合 K ( $K \subset V$ ) への到達可能性を問題とする S M T 信頼度(source-to-multiterminal reliability)問題として定式化される [24].

本章では、SMT信頼度問題の特別なクラスであるSAT信頼度(sourceto-all-terminal reliability)問題を取り扱う。SAT信頼度は、sからs以外のすべての頂点に至る正常な枝による経路が少なくとも1つ存在する確率として定義される。ただし、ここでは枝の故障は統計的に互いに独立なものとして扱うこととする[24]。

SAT信頼度表現式を導く算法のうち主なものを次に示す。

- (i) 繰返しIE法 (inclusion-exclusion method) [9].
- (ii) TR法 (topological reliability method) [24].
- (iii) SDP法 (sum of disjoint products method) [25].
- (iv) ファクタリング法 (factoring method) [27,28].
- (i)は、与えられたシステムが正常(あるいは異常)であるための極小な 枝集合としての木(あるいはカットセット)とそれらの可能なすべての組み合

せを考え、得られる集合のそれぞれを1つのブール積として信頼度表現式を導 くものである.したがって,このとき得られる表現式には積項の論理和および **論理減算を含み、互いに相殺される積項が数多く存在する.(i)で互いに相** 殺されない積項のみを求める算法として (ii) がある. これは p-acyclic sub graphs と呼ばれるアサイクリックグラフを求めることによって積項を得る算法 であり、p-acyclic subgraphs は、(i)で互いに相殺されない積項と1-1 対応している。(i),(ii)に対し,(iii)はシステムが正常(あるいは異 常)であるための極小な枝集合に対応するブール積項を列挙したのち、それら を互いに素な積項 (disjoint product) の和に展開するものである.この算法 では、演算過程で生成される中間的積項をすべて記憶する必要があり,また演 **算時間が展開の際の比較回数により大きく変動する.このとき,木の距離の短** い(あるいはカットセットの枝数が少ない)ものから演算することによって、 積項数および演算時間が削減されることは第2章で指摘した.また,一般的に グラフのカットセットの個数は木の個数よりも少ないため、SAT信頼度表現 式を導く算法として、カットセットを用いたSDP法が積項数および演算時間 において有効とされている[32]. 最近顕著な進展をみせている数値計算法であ る (iv) は、グラフ G の信頼度 R (G) が、枝 e を開放除去して得られるグラ フ (G-e) の信頼度 R(G-e) と、枝を短絡除去して得られるグラフ (G-e)\*e) の信頼度 R(G\*e) を用いて、次のように関係づけられることを利用 したものである.

 $R(G) = q_e R(G-e) + p_e R(G*e)$ ここで、 $p_e$ 、 $q_e$ はそれぞれ枝eが正常である確率、枝eが異常である確率を示す.

得られたグラフに対して、このファクタリング定理を繰り返し適用し、グラフが木か非連結グラフとなるまで続ける。この繰返し過程は二進木で表現でき、その葉の個数は少なくとも Gの木の個数に等しくなる。(G-e)、(G\*e)が共に連結グラフとなるように枝 e を選択するとき、その葉の個数が木の個数に等しい二進木構造が作られ、木の個数に比例する時間で演算することが可能となる[27,28]。

3.2節では本章で使用する記号等を定める.3.3節では枝の短絡・開放操作を基本とした木生成法[38,39,40,41,42],3.4節ではアサイクリックグラフの表現式の積項数がグラフの規模に関係なく1となることを利用したアサイクリックサブグラフ生成法[44],3.5節ではサイクルを含むグラフの強連結成分を1つの頂点に縮約して得られるグラフがアサイクリックグラフとなることを利用したセクショングラフ法[45],について述べる.最後に、計算機実験を通して各算法の比較検討を行う.

#### 3.2 諸定義

有向グラフG=(V,E) において、Vを始点Wを終点とする枝eをe=(V,W) で表す。本章で用いる、パス、連結、有向木などの用語はグラフ理論で通常に用いられるものであり[30]、定義を省略する。また、以下で対象とするグラフGを次にように定める。

- (i) ソース頂点sをただ1つ含む。
- (ii) 自己閉路は含まない。
- (iii) 連結である。
- (iv) sから残りの頂点に至るパス(有向木)が少なくとも1つ存在する。 Eの互いに素な部分集合  $E^-$ ,  $E^+$  ( $E^+$   $\cap$   $E^-$  =  $\phi$ ) に対して, $E^-$  の各枝を開放除去し,かつ  $E^+$  の各枝を短絡除去し,さらに短絡除去の過程で生ずる自己閉路も同時に開放除去することにより Gから得られるグラフを G ( $E^+$ ,  $E^-$ ) と書く.

Gの強連結成分を  $G_1=(V_1, E_1)$  ,  $G_2=(V_2, E_2)$  , …,  $G_c=(V_c, E_c)$  とし,各  $G_i$  (i=1, 2, …, c) をそれぞれ1 つの頂点  $v^*$  i に縮約して作られるグラフを G の縮約グラフといい,  $G^*=(V^*, E^*)$  と書く.本文で対象とするグラフ G は,  $G^*$  が連結でただ1 つの S を持つものに限るものとする.

Gのsを根とする有向木の集合を $\Upsilon$ で表し,Eの互いに素な部分集合 $E^+$ , $E^-$ に対し, $E^+$ を木枝としてもち, $E^-$ の枝を含まない $\Upsilon$ の部分集合を $\Upsilon$ ( $E^+$ , $E^-$ )で表す.

グラフ Gの任意の木  $T_i \in \mathcal{T}$ の枝集合を  $E(T_i)$  で表す.

$$E(T_i) - E(T_j) = \{e_{i1}, e_{i2}, \dots, e_{ir}\}$$
 (3-1)

$$E(T_{j}) - E(T_{i}) = \{e_{j1}, e_{j2}, \dots, e_{jr}\}$$
 (3-2)

であるとき,

$$T_{j} = \frac{\{e_{j1}, e_{j2}, \cdots, e_{jr}\}}{\{e_{i1}, e_{i2}, \cdots, e_{ir}\}} T_{i}$$
 (3-3)

と書き,  $r \in T_i \subset T_i$ の距離という.

頂点集合  $U \subseteq V$ に関するセクショングラフ  $G(U) \triangleq (U, (U \times U) \cap E)$ から作られる縮約グラフを  $G^*(U^*)$  で表す。また、記述を簡単にするため以下の記号を定める。

 $x_e(\bar{x}_e)$ : 枝 e が正常 (異常) であることを示すブール変数.

| *E* | : 集合 *E* に含まれる要素数.

 $N:1 \sim |V|$ までの連続する整数集合.

#in∈ N: Nに含まれる最小の整数. #in=1.

 $n_c \in N: N$ に含まれる最大の整数.  $n_c = |V|$ .

 $V_{i}$ :  $v^{*}$ ,に対応する Gの強連結成分の頂点集合.

 $N_i: min_i = n_{Gi-1} + 1 \text{ is } n_{Gi} = min_{i+1} - 1 \text{ } (n_{Gi} - min_i + 1 = |V_i|)$ 

までの連続する整数集合. ただし,  $n_{co}=0$ ,  $m_{in_{c+1}}=|V|+1$ .

なお、cはGに含まれる強連結成分の数を示す...

α:グラフの頂点の位相的順序 (topological order)

$$\alpha : V^* \to I, I = \{1, 2, \dots, |V^*|\}$$
.

 $E_{r}(v): v$ を共通の終点とする枝集合.

 $(V_1, V_2)$ : 互いに素な頂点集合  $V_1, V_2$ に関する枝集合  $\{(u, v) \in E \mid u \in V_1, v \in V_2\}$ .

 $P_{\nu}(\nu)$ :  $\nu$ が属する強連結成分の頂点集合.

 $P_N(v)$ :頂点集合  $P_v(v)$  に対応する整数集合.

# 3.3 木生成法

本節では、得られる表現式の良さを定める目的関数を積項数におき、最適な表現式を得るために極小集合が具備すべき条件およびその処理順序について考察し、極小集合の個数に等しい積項数を持つ表現式導出のための条件を示す。また、枝の短絡・開放除去を基本とした木の生成法を述べ、ブール演算を行うことなく、木の総数に等しい積項数を持つSAT信頼度表現式を導く算法を提案する。なお、最適な表現式を導く動機およびその過程は異なるものの、本節で得られる表現式は文献[27]~[29]で示されるファクタリング法による数値計算法を修正しても導ける。極小枝集合として木を用いることは、得られる積項数が木の個数に制限される欠点を持つが、木生成法は木に対応するブール積項間の比較が必要でないため高速であり、またカットセットを用いたSDP法よりも使用するメモリが遙かに少なくなるなどの特徴を持っている。

## 3.3.1 信頼度表現式

SAT信頼度問題において、システムがsを根とする正常な枝だけから成る有向木を少なくとも1つ含むならば、そのシステムは正常とみなされ、またsと1つ以上の頂点を分離するような故障枝から成るカットセットを少なくとも1つ含むならば、そのシステムは異常とみなされる。これら極小集合(木あるいはカットセット)をすべて生成すれば、SAT信頼度(非信頼度)表現式を導くことができる[24]。

i番目の木(カットセット)に対応するブール変数の積を  $A_i$ とおけば、システム信頼度(非信頼度表現式)Fが次式で表せる[15]。

$$F = A_1 + A_2 + \cdots + A_m \tag{3-4}$$

ここで、mは極小集合の総数を示す。また、極小集合が木に相当するときA,は木に属する枝 e に対応するブール変数 x の積を、カットセットに相当するとき A,はカットセットに属する枝 e に対応したブール変数  $\overline{x}$  の積をとる。

一般に、各 $A_i$  (i=1, 2, …, m) は互いに素 (disjoint,  $A_k \cdot A_j = 0$ )

とはならないため、式(3-4)は信頼度計算式には対応しない。換言すれば、x。に枝 e の信頼確率を、 $\bar{x}$ 。に枝 e の非信頼確率を代入しても、得られる値がシステムの信頼度あるいは非信頼度とはならないことを意味している。

このためSDP法では式(3-4)に対して、次のような展開が行われる.

 $F = A_1 + \bar{A_1} A_2 + \bar{A_1} \bar{A_2} A_3$ 

$$+ \cdots + \overline{A}_{1} \overline{A}_{2} \cdots \overline{A}_{j} \cdots \overline{A}_{i-1} A_{i}$$

$$+ \cdots + \overline{A}_{1} \overline{A}_{2} \cdots \overline{A}_{m-1} A_{m}$$

$$(3-5)$$

式(3-5)の i 番目の項

$$F_i = \overline{A}_1 \, \overline{A}_2 \, \cdots \, \overline{A}_j \, \cdots \, \overline{A}_{i-1} \, A_i \tag{3-6}$$

の演算において、 $A_J$ にあって $A_J$ にないブール変数の集合を $B_{JJ}$ 、その積を $A_{JJ}$ とおけば、 $F_J$ は次のように書き換えることができる[31].

$$F_i = \overline{A}_{1i} \overline{A}_{2i} \cdots \overline{A}_{ji} \cdots \overline{A}_{i-1,i} A_i$$
 (3-7)

また、式(3-7)において、任意の2つの積項 $A_{ji}$ ,  $A_{ki}$ が $B_{ji}$ つ  $B_{ki}$ の関係を持つとき、式(3-7)から $\overline{A}_{ji}$ を除去することができる.したがって、除去の結果残った積項を $\overline{A}_{ji}$  (j=1, 2, …, b) とおけば、 $F_{i}$ として次式が得られる.

$$F_{i} = \overline{A}'_{1i} \overline{A}'_{2i} \cdots \overline{A}'_{ji} \cdots \overline{A}'_{bi} A_{i}$$

$$(3-8)$$

式(3-8)の $\overline{A}'_{j,i}$ は、 $A'_{j,i}=x_1x_2$  …  $x_k$ とおくとき、次のような互いに素な積項の和に展開される。

$$\bar{A}'_{ji} = \bar{X}_1 + X_1 \bar{X}_2 + X_1 X_2 \bar{X}_3 + \cdots + X_1 X_2 \cdots X_{k-1} \bar{X}_k$$
 (3-9)

#### 3.3.2 表現式の最適化

SDP法によって導かれる表現式の良さを定める目的関数を積項数におくならば、与えられた m個の A に対して、最適な表現式はその積項数が m より小さくない最小数となるものと定義できるであろう。もし m 個のものが得られるとすれば、それらはどのような条件のとき得られるかを調べ、その演算法を確立することは興味ある問題である。ここで、SAT信頼度問題以外でこのような場合の一例を挙げておく。

[例題3-1 ] [25] 2-out-of-4:G システムにおいて\*, その構成要素およびそのブール変数表現をA, B, C, Dで表せば, A, (i=1,2,3,4,5,6) として, AB, AC, BC, BD, AD, CDが得られる. これに対して式(3-5)を適用すれば, システムの信頼度表現式 $F_R$ が次式のように得られる.

 $F_R = AB + \bar{B}AC + \bar{A}BC + \bar{A}\bar{C}BD + \bar{B}\bar{C}AD + \bar{A}\bar{B}CD$  (3-10) 一般に、積項数の少ない表現式を得るには、

- (i) 式(3-4)に対する m個の A;の処理順序,
- (ii) 式(3-8)に対する b 個の A / i の処理順序

をそれぞれ独立に考察する必要がある。これまで(i)の観点からの考察がいくつかなされている。カットセットの順序づけについては,それに属する枝数の少ないものを優先する算法[32],木の順序づけについては,隣り合った木の間の距離をつねに1に保つ算法(例題3-1の場合はこれに類似している) [26],が有利であることが確かめられている。しかし,(ii)の観点からの組織的な削減法を加味したものは見あたらない。

積項数が最小個数のmとなるためには,各 $F_i$ ( $i=2, \cdots, m$ )がただ1つの積項からなる必要がある。以下では,これが成立するために, $B_{ji}$ (あるいは $A_{ji}$ )が具備すべき条件とその処理順序について考察する.

ある $oldsymbol{A}_{II}$ と $oldsymbol{A}_{II}$ の積 $oldsymbol{A}_{II}$ の有 $oldsymbol{A}_{II}$ の有 $oldsymbol{A}_{II}$ の有 $oldsymbol{A}_{II}$ のの表示な元の個数が $oldsymbol{1}$ となる集合を

$$B' = \{ B'_{1i}, B'_{2i}, \cdots, B'_{ti} \}, t < b,$$
(3-11)

対応する変数の集合を

$$X' = \{ x_{1i}, x_{2i}, \dots, x_{ti} \}$$
 (3-12)

とおくと, 次式が成立する.

 $P_i = \overline{A}'_{1i} \overline{A}'_{2i} \cdots \overline{A}'_{ti} A_i$ 

<sup>\*</sup> システムに含まれる4個の要素のうち、少なくとも2個の要素が正常ならば、システムが正常と成るものをいう.詳細は第4章に記述している.

$$= \overline{X}_{1,i} \, \overline{X}_{2,i} \, \cdots \, \overline{X}_{t,i} \, A_{i} \tag{3-13}$$

つぎに、 | *B*<sub>ji</sub> | ≥ 2 である集合を

$$B = \{ B_{1i}^{"}, B_{2i}^{"}, \cdots, B_{ri}^{"} \}, r = b - t$$
 (3-14)

とするとき,B "の任意の要素,例えばB " $_{II}$ に対して $\overline{A}$ " $_{II}$ ・P  $_{I}$ がただ1 つの積項

$$\overline{A}^{"}_{ij} \cdot P_i = P_i \tag{3-15}$$

となる条件は、式(3-9)より次の関係を満たすことである.

$$B \stackrel{"}{}_{ii} \cap X \stackrel{\prime}{\neq} \phi \tag{3-16}$$

このことは次の2つの式が成立することから明かである.

$$\bar{\mathbf{x}} \cdot P_i = P_i \tag{3-17}$$

$$x \cdot \overline{A}^{"'}_{ii} \cdot P_i = 0 \tag{3-18}$$

ここで,  $x \in B''_{ji} \cap X'$ ,  $B'''_{ji} = B''_{ji} - \{x\}$ , 0 は零元を示す.

一般に、 $A_{JJ}$ の演算順序は変数の少ないものを優先した方がよいといわれている[15]. しかし、上述の条件を満たす場合は、 $B_{J}$ に属するものを任意の順に処理し、 $B_{J}$ によるものを省略してもよいことが明らかである.

#### 3.3.3 木の生成

与えられたグラフG = (V, E) に含まれる木のすべてを重複なく生成する算法は、これまで数多く報告されているが、計算複雑度の観点からは $O(|V| + |E| + m \cdot |E|)$  のものが高速である[35]. ここではこのようなもののうち、枝の短絡・開放操作を基本としたものを若干修正して述べ[36,37]、表現式の最適化が可能な生成法であることを示す.

Gのソースから出る枝  $e_1$  = (s, v) を用いて,Gの木集合 $\mathcal{T}$ は  $\mathcal{T}$  =  $\mathcal{T}$  ( $\{e_1\}$ ,  $\phi$ ) +  $\mathcal{T}$  ( $\phi$ ,  $\{e_1\}$ ) (3-19) と書くことができる.ここで(+)は直和を表す.

 $\mathcal{F}(\{e_1\}, \phi)$  と  $\mathcal{F}(\{e_1\}, \phi)$  の木の間, $\mathcal{F}(\{e_1\}, \phi)$  の木の間にはそれぞれ 1-1 対応がある。  $\mathcal{F}(\{e_1\}, \phi)$  のソース(以下では super source vertex という)から出る枝  $e_2$ に対して

$$\mathcal{F}(\lbrace e_1 \rbrace, \phi)$$
  
= $\mathcal{F}(\lbrace e_1, e_2 \rbrace, \phi) + \mathcal{F}(\lbrace e_1 \rbrace, \lbrace e_2 \rbrace)$  (3-20)  
と書ける、

一般に、 $\mathcal{T}$  ( $E^+$ ,  $E^-$ ) はG ( $E^+$ ,  $E^-$ ) の super source vertex から出る枝集合  $B=\{e_a, e_b, e_c, \cdots, e_x\}$  を用いて

$$\mathcal{T}$$
 (  $E^+$ ,  $E^-$ )
$$= \mathcal{T}$$
 (  $E^+$ + {  $e_a$  } ,  $E^-$ )
$$+ \mathcal{T}$$
 (  $E^+$ ,  $E^-$ + {  $e_a$  } )
$$= \mathcal{T}$$
 (  $E^+$ + {  $e_a$  } ,  $E^-$ )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_b$  } ,  $E^-$ + {  $e_a$  } )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_a$  } ,  $e_b$  )
$$\vdots$$

$$= \mathcal{T}$$
 (  $E^+$ + {  $e_a$  } ,  $E^-$ )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_a$  } ,  $E^-$ + {  $e_a$  } )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_b$  } ,  $E^-$ + {  $e_a$  ,  $e_b$  } )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_b$  } ,  $E^-$ + {  $e_a$  ,  $e_b$  } )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_b$  } ,  $E^-$ + {  $e_a$  ,  $e_b$  } )
$$+ \mathcal{T}$$
 (  $E^+$ + {  $e_b$  } ,  $E^-$ + {  $e_a$  ,  $e_b$  } )

と書くことができる.

図3-1に示すアルゴリズムは、これらを ALGOL-like な言語で記述したものである。

式(3-21)の右辺の各項は

$$G (E^{+} + \{e_{a}\}, E^{-}),$$
 $G (E^{+} + \{e_{b}\}, E^{-} + \{e_{a}\}),$ 
 $G (E^{+} + \{e_{c}\}, E^{-} + \{e_{a}, e_{b}\}),$ 
 $\vdots$ 
 $G (E^{+} + \{e_{x}\}, E^{-} + B - \{e_{x}\})$ 

の木集合に対応する.これらのグラフのおのおのに1つの頂点 $g_s$ ,  $g_s$ ,  $g_s$ ,  $g_s$ , ...,  $g_x$ を割り当て,それらの頂点を $G(E^+, E^-)$ に対応する頂点 $g_s$ の子とし,枝ラベル $g_s$ ,  $g_s$ ,  $g_s$ , ...,  $g_s$ ,  $g_s$ ,  $g_s$ , ...,  $g_s$ ,  $g_s$ ,  $g_s$ ,  $g_s$ , ...,  $g_s$ ,  $g_s$ , g

```
procedure SPANNING TREE
   begin
      procedure GROW(G(E*,E*))
          begin
             comment: let e and B be variables local to procedure CROW;
             let B be a set of edges directed away
                   from super source vertex of G(E^+,E^-);
             if |E^+| = |V| - 1 then output E^+ and E^-;
             else
                begin
                    while B≠ø do
                       begin
                          select an edge e & B and B:=B-{e};
                          GROW(G(E+\{e\},E^-));
                          E^{-}:=E^{-}+\{e\}
                       end
                 end
          end CIROW;
       if a condensed graph G' is connected
             and contains only one source
          then GROW(G(\emptyset,\emptyset));
          else output 'no trees'
   end SPANNING TREE;
```

図3-1 木生成アルゴリズム

ものは後続するものの兄と呼ぶことにする。この関係を右側優先で図示したのが図3-2である。したがって、 $E^-$ に加えられる枝集合は、自分の兄(右側の頂点)に接続される枝ラベルの集合とみることができる。

図3-1のアルゴリズムを図3-2で定義される生成木(generation tree)上で述べると、この生成木を深さ優先探索するもので、ある葉に到達するとき 1 つの木が生成されることになり、そのとき  $E^+$ は生成木の根から葉に至る枝ラベルの集合となる。またある頂点(たとえば図3-2のg) 以降がすべて探索されれば自分自身の弟(たとえばg) へ、またもし自分の弟がなければ、親の弟へと探索が進められる。なお、ある頂点(たとえばg) に対応するグラフを $G(E^+$ ,  $E^-$ , とすると、 $E^+$ , は根からg, に至るパス上にある枝ラベルの集合に対応し、 $E^-$ , はg, の兄にはいる枝ラベルの集合と、g, の祖先の兄にはいる枝ラベルの集合の和に対応することになる。

以下では、i番目に生成された木を $T_i$ 、対応するグラフを $G(E^+_i, E^-_i)$ と書くことにする。後述するが、この $E^-_i$ は信頼度表現式の導出に重要な役割を果たすものである。

[例題3-2] 図3-3のグラフにアルゴリズムを適用した結果を図3-4の生成木で示す。なお無向枝は互いに逆向きの2本の有向枝として取り扱っている。また図3-5は図3-4の生成木から得られる木T,と $E^-$ ,を描いたものである。

いま,得られた木を $T_1$ ,  $T_2$ , …,  $T_i$ , …,  $T_m$ と並べるとき,木 $T_i$  (2  $\leq i \leq m$ ) と木集合

$$\mathcal{T}_{i-1} \triangleq \{ T_1, T_2, \cdots, T_{i-1} \}$$
 (3-22)

の間には次のような関係が成立する.

(A) 任意の枝  $e \in E^{-}$ ,に対して、次の関係を満たす距離 1 の木 T  $_{\prime} \in \mathcal{T}$   $_{\prime-1}$  が存在する.

$$E^{+}_{j} - E^{+}_{i} = \{e\}$$
 (3-23)

(B) どの木  $T_i \in \mathcal{T}_{i-1}$ についても,次式が成立する.

$$E^{+}{}_{j}\cap E^{-}{}_{i}\neq \emptyset \tag{3-24}$$

(B)については、生成木上の根から葉T,に至るパス上の各頂点の兄はすべて探索ずみであること、パス上からこれらの頂点を結ぶ枝ラベルの集合が $E^-$ で

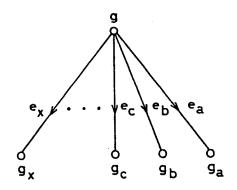


図3-2 式(3-21)の図的表現

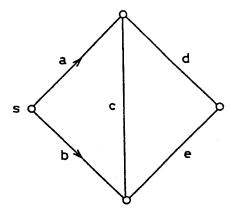


図3-3 グラフ G

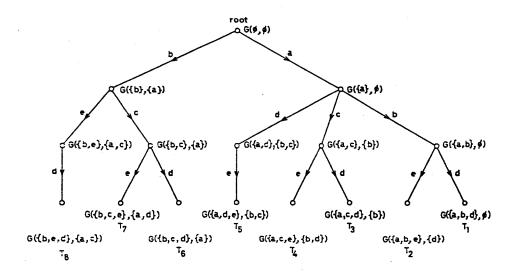


図3-4 図3-3の生成木

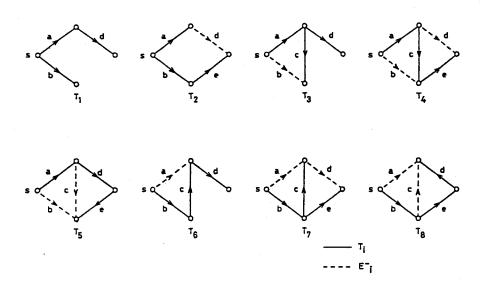


図3-5 木 T,と E-,の関係

あることを考慮すれば明らかであろう.

次に (A) について示そう。e の終点をその終点とする枝 e  $_{i}$   $\in$  E  $(T_{i})$  が存在するので,常に距離 1 の  $T_{i}$  が次のようにして得られる。

$$T_{j} = \frac{\lbrace e \rbrace}{\lbrace e_{i} \rbrace} \quad T_{i} \tag{3-25}$$

いま,j>iと仮定しよう.グラフG ( $E^+_{i1}$ ,  $E^-_{i1}$ ) およびG ( $E^+_{i1}$ +  $\{e_i\}$  , $E^-_{i1}$ ) に対応する生成木上の頂点をそれぞれ $g_1$  , $g_2$ とおけば,枝ラベルe は $g_1$ から葉 $T_i$ に至るパス上に現れることになる.また,枝eと $e_i$ の終点が同一であることから,G ( $E^+_{i1}$ +  $\{e_i\}$  , $E^-_{i1}$ ) とそれ以降の短絡除去操作によって得られるどのグラフの枝集合Bにおいてもe  $\in$  Bが成立する.すなわち,生成木上の根から $g_2$ を経由して葉に至るパスによって定められるどの木 $T_k$  $\in$  T ( $E^+_{i1}$ +  $\{e_i\}$  , $E^-_{i1}$ ) においても,常にe  $\in$  E ( $T_k$ ) が成立する.このことは,根から $g_1$ に至るパス上の各頂点の兄を終点とする枝ラベルの1つとしてe が割り当てられていること,つまり探索ずみであることを意味し,e  $\in$   $E^-_{i1}$   $\in$   $E^-_{in}$ が成立する.よって,e  $\in$  E ( $T_i$ ) を満たす  $T_i$ は存在しないこととなり矛盾する.したがって,i < i が成立する.

上述の関係(A), (B) はそれぞれ式(3-11), (3-16)に対応しており, 3. 3. 2節の後半において議論された積項数 m の表現式導出のための条件等を満たす木の生成が可能であることを示している。さらに, 木の生成に付随して得られる  $E^-$ , が変数の集合 X 'に対応するため, 式(3-7)の式(3-9)による展開(ブール演算)を行うことなく, 式(3-13)が直接導かれる。すなわち, i 番目の項  $F_i$  は次式のように展開される。

$$F_{i} = \overline{A}_{1} \overline{A}_{2} \cdots \overline{A}_{i-1} A_{i}$$

$$= \prod_{e \in E^{-}_{i}} \overline{X}_{e} \cdot \prod_{e \in E^{+}_{i}} X_{e}$$

$$(3-26)$$

ここで、∏はブール変数の論理積演算を示す.

図3-3の $F_R$ が図3-5の $T_i$ (図3-4の $E^+_i$ )と $E^-_i$ から次のように導かれる。 $F_R = X_a X_b X_d + \overline{X}_d X_a X_b X_e + \overline{X}_b X_a X_c X_d + \overline{X}_b \overline{X}_d X_a X_c X_e + \overline{X}_b \overline{X}_c X_a X_d X_e + \overline{X}_a X_b X_c X_d + \overline{X}_a \overline{X}_d X_b X_c X_e$ 

## 3.4 アサイクリックサブグラフ生成法

枝の短絡・開放操作を基本とした木の生成法を用いることにより、木の総数に等しい積項数を持つSAT信頼度表現式を導く算法を3.3節に提案した。しかしながら、グラフに含まれる木の個数がグラフの規模とともに指数的に増加するため、得られた信頼度表現式を用いて信頼度を算出できるグラフは小さなものに限られることになる(例えば、頂点数 |V| の完全グラフでは木の個数は |V| できるなる)。

サイクルを含まない有向グラフのどの枝(i, j)  $\in E$ も i < j となるように頂点に整数を割り当てること(頂点の位相的順序,topological order)ができる。この順序にしたがい枝(あるいは並列枝)を短絡除去して得られる信頼度表現式の積項数はグラフの規模に関わりなく 1 となる。

信頼度表現式が与えられるとき、信頼度は各枝の信頼確率を対応するブール変数に代入することにより求められるので、演算回数は信頼度表現式に含まれる演算記号数に等しくなる。したがって、本節では得られる表現式の良さを定める目的関数を表現式の大きさ(リテラル(literal)数、あるいは演算記号(operator、+あるいは×)数)におく。

本節では、一般の有向グラフに対して枝の短絡・開放操作を行うことにより、サイクルを含まないサブグラフ(アサイクリックサブグラフ)を生成し、信頼 度表現式がアサイクリックサブグラフに対応する積項の和として得られること を示す.

#### 3.4.1 既知の結果

図3-6のサイクルを含まないグラフG。に対して3.3節のアルゴリズムを適用すれば、図3-7および表3-1が得られる。図3-7はsを根(root)とする4個の有向木(arborescence)を深さ優先探索(depth-first search)により生成し

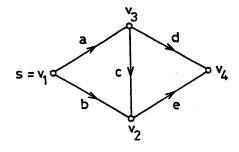


図3-6 サイクルを含まないグラフ G。

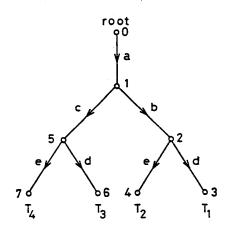


図3-7 有向木生成のための探索木

表3-1 有向木と弦

node	E+	E-	Ti
0	ø	ø	
1	а	ø	
2	a, b	ø	
3	a, b, d	ø	$T_1$
4	a, b, e	d	T <sub>2</sub>
5	a, c	b	
6	a, c, d	Ь	$T_3$
7	a, c, e	b, d	$T_4$

たものである。図3-7において,根から葉(leaf)に至る枝 > ベルが 1 つの有向木を示している。表3-1において,これらの有向木に関するそれぞれの弦(chord)を $E^-$ で示している。表3-1から,G。の信頼度表現式F (G。)を次のように導くことができる。

$$F (G_s) = X_s X_b X_d + \overline{X}_d X_s X_b X_s$$

$$+ \overline{X}_b X_s X_c X_d + \overline{X}_b \overline{X}_d X_s X_c X_s \qquad (3-28)$$

式(3-28)を因数分解すれば、次式が得られる。

$$F (G_s) = X_s (X_b + \bar{X}_b X_c) (X_d + \bar{X}_d X_c)$$
 (3-29)

次式を用いれば、

$$X_{b} + \overline{X}_{b} X_{c} = X_{b} + X_{c} \tag{3-30}$$

式(3-29)は次のように書き換えることができる.

$$F (G_a) = X_a (X_b + X_c) (X_d + X_e)$$
 (3-31)

サイクルを含まない有向グラフGのソースsを除くどの頂点( $V-\{s\}$ )

においても、それを終点とする有向枝が少なくとも 1 本存在すれば、Gが有向木を持つことは明らかである。いま、頂点 v を共通の終点とする枝集合を E 、(v) で表せば、頂点 v を終点とする正常な有向枝が少なくとも 1 本存在する事象を表す信頼度表現が次のように示せる [43] .

$$e \in \sum_{t \in E_{t}(v)} X_{e}$$

したがって、グラフGの信頼度表現式Fは次式で表わせる.

$$F = \prod_{v \in V - \{s\}} \sum_{e \in E_t(v)} x_e \tag{3-32}$$

式(3-32)を図3-6のグラフG。に適用すれば、信頼度表現式F(G) が次のように直接導ける.

$$F (G_b) = X_b (X_b + X_c) (X_d + X_c)$$
 (3-33)

式(3-33)は式(3-31)と一致している。そこで、式(3-30)を用いて式(3-33)を書き換えれば、図3-7の探索木 (rooted search tree)を生成することなく式(3-29)が得られる。

サイクルを含まないグラフのどの枝(i, j) $\in E$ も、i<jとなるように頂点に整数を割り当てることができる。図3-6の頂点は次のような順序づけ $\alpha$ が可能である。

$$\alpha$$
 (s) = 1,  $\alpha$  ( $v_3$ ) = 2,

$$\alpha$$
 ( $v_2$ ) = 3,  $\alpha$  ( $v_4$ ) = 4

また、G。の頂点をそれぞれ終点とする枝集合が次のように示せる.

$$E_t(s) = \phi$$

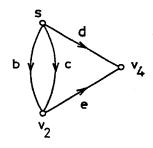
$$E_{t}(v_{3}) = \{a\}$$

$$E_{t}(v_{2}) = \{b, c\}$$

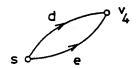
$$E_{t}(v_{A}) = \{d, e\}$$

この順序に従ってG。から生成される探索木を示そう(図3-8, 3-9参照).

 $G_s$ において、ソースから出る枝  $a=(s, v_s)$  を短絡除去(s と  $v_s$  を重ね合わせ、枝 a を除去)すれば、図3-8(a)に示すグラフ  $G_s$ ( $\{a\}$ ,  $\phi$ )が得られる。 $G_s$ ( $\{a\}$ ,  $\phi$ )において、s から  $v_s$  への並列枝 b, c が存在する。並列枝 b, c を同時に短絡除去( $b \parallel c$  と書き、b, あるいは c, あるいは b



(a) G<sub>a</sub>({a}, Ø)



(b)  $G_a(\{a, b || c\}, \emptyset)$ 

図3-8 G。を短絡操作して得られるグラフ

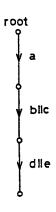


図3-9 G.の探索木

および c を短絡除去することを意味する)すれば, $G_{a}$  ( $\{a, b \mid c\}$ ,  $\phi$ ) が得られる(図3-8(b)参照). さらに,並列枝短絡除去操作  $d \mid e$  を行うことにより  $G_{a}$  ( $\{a, b \mid c, d \mid e\}$ ,  $\phi$ ) が得られる.以上の操作から図3-9 の探索木が生成でき,またこの探索木から式(3-33)が直接導ける.サイクルを含まないグラフのすべての頂点に対して位相的順序で整数を割り当てることができるため,グラフの規模に関わらず葉の個数は1となり,また得られる表現式の積項数も1となる.

# 3.4.2 アサイクリックサブグラフの生成

有向グラフはサイクルを含まないものよりサイクルを含むグラフが一般的である。 図3-10のグラフ  $G_c$ の表現式の導出について考える。

グラフ $G_c$ はサイクルを含むので,すべての頂点を順序づけることは不可能である.グラフ $G_c$ に対して,強連結成分を1つの頂点として得られる縮約グラフ $G_c$ \*を考える(図3-11参照).縮約グラフ $G_c$ \*はアサイクリックグラフであり,その頂点は次のような順序づけが可能である.

$$\alpha (s) = 1$$
,  $\alpha (v_2) = 2$ ,  
 $\alpha (v_3) = 3$ ,  $\alpha (v_4) = 4$ 

図3-10, 3-11から,  $v^*$ ; (i=1, 2, 3, 4) に対応する G 。の頂点集合 V; および V; の各頂点に割り当て可能な整数集合 N; が次のように示せる.

$$V_{1} = \{ s \}$$
,  $N_{1} = \{ 1 \}$   
 $V_{2} = \{ v_{2}, v_{3}, v_{4} \}$ ,  $N_{2} = \{ 2, 3, 4 \}$   
 $V_{3} = \{ v_{5}, v_{6}, v_{7} \}$ ,  $N_{3} = \{ 5, 6, 7 \}$   
 $V_{4} = \{ v_{8} \}$ ,  $N_{4} = \{ 8 \}$ 

枝  $b=(s, v_3)$ ,  $c=(s, v_2)$  の存在から、整数 2 を割り当て可能な頂点が  $v_2$  あるいは  $v_3$  に限られることは明かである.

G。から枝 b を短絡除去して得られるグラフを G。( $\{b\}$ ,  $\phi$ ),枝 b を開放除去して得られるグラフを G。( $\phi$ ,  $\{b\}$ ) とおくとき,G。の信頼度表現式 F (G。) は次のように導ける.

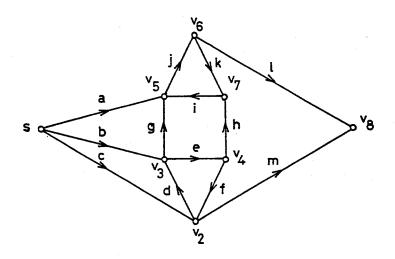


図3-10 サイクルを含むグラフ G。

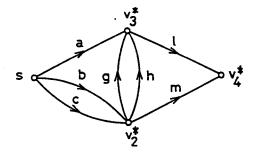


図3-11 縮約グラフ G 。\*

$$F (G_c) = x_b F (G_c (\{b\}, \phi)) + \bar{X}_b F (G_c (\phi, \{b\}))$$
 (3-34)

式(3-34)はネットワーク信頼度におけるファクタリング理論を表す式として知られている[28]. この式を利用してアサイクリックサブグラフを生成することを考える.

いま、 $v_3$ に整数 2 を割り当て枝 b を短絡除去すれば、グラフ  $G_c$  ( $\{b\}$ ,  $\phi$ ) が得られる。 $G_c$  ( $\{b\}$ ,  $\phi$ ) の頂点集合  $V_2 = \{v_2, s, v_4\}$  に関する部分図を図3-12(a)に示す。図3-12において、 $\bigoplus$  の頂点はそれぞれ super source vertex を示す。また、図3-12(a)において、枝  $d=(v_2, s)$  のどのような状態も  $G_c$  ( $\{b\}$ ,  $\phi$ ) の信頼度に影響を及ぼすことはなく、枝 d を  $G_c$  ( $\{b\}$ ,  $\phi$ ) から開放除去できる(枝 d を点線で示す)。 $G_c$  ( $\{b\}$ ,  $\phi$ ) から枝 d を開放除去すれば、 $v_2$ 、 $v_4$  がサイクル上の頂点でなくなるため 次のような整数の割り当てが可能である。

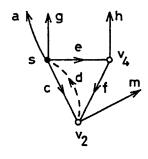
 $\alpha (v_2) = 4, \alpha (v_4) = 3$ 

この順序にしたがって枝  $e=(s, v_4)$  を短絡除去すれば, $G_c(\{b, e\}, \phi)$  が得られる(図3-12(b)参照). さらに, $G_c(\{b, e\}, \phi)$  において,並列枝  $c=(s, v_2)$  ,  $f=(s, v_2)$  を同時に短絡除去すれば, $G_c(\{b, e\}, c \mid\mid f\}$  , $\phi$ )が得られる(図3-12(c)参照).

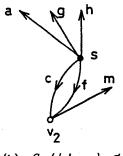
ここで、アサイクリックサブグラフ生成法とファクタリング法との相違を枝の短絡・開放操作に関して示しておこう.

(i) 図3-10が与えられるとき,ファクタリング法ではsから出る枝a,b,cに対する短絡・開放操作の順序に区別は無いが,アサイクリックサブグラフ法では頂点に対する位相的順序の関係から初期のレベルでの操作対象は枝b,cのみであり,枝aはその後のレベルでの操作対象となる.これに反して,同一のレベルにおいて,a,b,cの順序で枝の短絡・開放操作を行えば,表現式が次のように書ける.

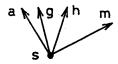
$$F (G_c) = x_b F (G_c (\{a\}, \phi))$$
  
  $+ \overline{x}_b x_b F (G_c (\{b\}, \{a\}))$ 



(a)  $G_c(\{b\}, \emptyset)$ 



(b) G c({b, e}, ∅)



(c)  $G_c(\{b, e, c || f\}, \emptyset)$ 

図3-12 G。の枝を短絡除去して得られるグラフの部分図

$$+ \bar{x}_{s} \bar{x}_{b} x_{c} F (G_{c} (\{c\}, \{a, b\}))$$
(3-35)

式(3-35)の表現式から得られる積項数は6である.なお、提案算法から得られる表現式の積項数は4である(後述の式(3-37)参照).また、同一のレベルにおいて、b, c, a の順序に枝を短絡・開放操作すれば、その表現式は次のように書ける.

$$F (G_c) = X_b F (G_c (\{b\}, \phi)) + \bar{X}_b X_c F (G_c (\{c\}, \{b\})) + \bar{X}_b \bar{X}_c X_b F (G_c (\{a\}, \{b, c\}))$$
(3-36)

しかしながら、 $G_c$ ( $\{a\}$  、 $\{b,c\}$  )のソースsから頂点 $v_2$ 、 $v_3$ 、 $v_4$ に至る経路が存在しないため、第3項の信頼度は0となる。アサイクリックサブグラフ法では、頂点に対する位相的順序を用いることにより積項数の少ない(またリテラル数の少ない)表現式が生成でき、また不要な操作が排除できるため表現式の生成が高速である。

(ii) ファクタリング法では図3-12(a)における枝 e , 図3-12(b)における並列枝 c , f に対する開放除去操作が行われるが,アサイクリックサブグラフ法では頂点  $v_2$  ,  $v_4$  に対する操作順序が上述のように決まるため,これらの枝に対する開放除去操作は行われない.なぜなら, $G_c(\{b\}, \{e\})$  ,  $G_c(\{b, e\}, \{c \mid f\})$  のいずれのグラフの信頼度も0となるからである.アサイクリックサブグラフ法では頂点に対する位相的順序により不要な操作が排除できるため,表現式の導出時間を減少できる.

以上の議論から,図3-13に示すアルゴリズム REL\_EXPRESSION を構築できる. アルゴリズム REL\_EXPRESSION を図3-10に適用するとき,得られる生成木の頂 点は $G_c$ から枝を短絡・開放操作して得られるグラフ $G_c$  ( $E^+_i$ ,  $E^-_i$ ) に対応 している.図3-13において, $B_i$  ( $i \neq n_c$ ) はi番目に短絡・開放操作される 枝あるいは並列枝の候補である.また, $B_i = \emptyset$ であるとき,グラフ $G_c$  ( $E^+_i$ ,  $E^-_i$ ) の縮約グラフを作成し,得られる $V_i$ の関係から $B_i$ を求めている.

```
procedure REL EXPRESSION(G=(V,E))
   begin
        procedure GROW(G(E^+,E^-),G[V],N,i,l,F_R)
            comment: G(V) is a vertex-section subgraph of G(E^*, E^-);
            begin
                 if i = |V| then output F_R;
                 else
                     begin
                          delete arcs directed toward a super source vertex s;
                          if B<sub>i</sub> is unknown then
                              begin
                                   l := l + 1;
                                  construct the condensed graph G^*(V^*);
                                  find a topological order \alpha;
                                   determine the partitions of V and N pertinent to \alpha:
                                       \{V_1, V_2, \dots, V_c\}, \{N_1, N_2, \dots, N_c\};
                                  form level, l's arc sets
                                  B_{\mathbf{n}_k} \triangleq (\sum_{j=1}^k V_j, V_{k+1}), k=1,2,\cdots,c-1;
let B_j (level l_i's arc sets, l_i < l and i < j) be known
                              end:
                          while B<sub>i</sub>≠ø do
                              begin
                                   select an arc e=(s,v) \in B_i;
                                   B_i := B_i - \{e\};
                                   GROW(G(E^{+}\cup \{e\}, E^{-}), G[Pv(v)], Pv(v), i+1, l, x_eF_R);
                                   E^{-}:=E^{-}\cup\{e\};
                                   F_R := \overline{x}_e F_R;
                                   comment: if e is a multiple arc consisting of
                                                e_1,e_2,e_3,\cdots, we write E \cup \{e_1,e_2,e_3,\cdots\},
                                                E^+ \cup \{e_1 \| e_2 \| e_3 \| \cdot \cdot \cdot \}, \quad x_e = x_{e_1} x_{e_2} x_{e_3} \cdot \cdot \cdot ,
                                                x_e = x_{e_1} + \overline{x}_{e_1} x_{e_2} + \overline{x}_{e_1} \overline{x}_{e_2} x_{e_3} + \cdots;
                              end
                          let B_i be unknown
                      end
             end GROW;
         GROW(G(\phi,\phi),G(V,E),N,1,0,1)
     end REL EXPRESSION;
```

図3-13 アルゴリズム REL\_EXPRESSION

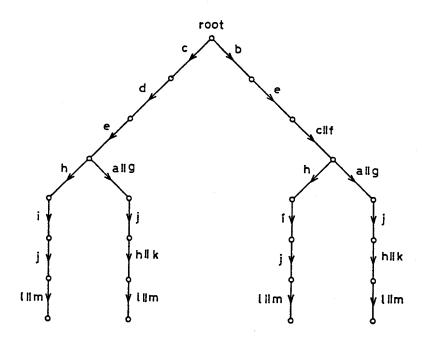


図3-14 G。の探索木

図3-10にアルゴリズム REL\_EXPRESSION を適用すれば、図3-14に示す(右側優先の)深さ優先探索により作成された探索木を生成することができる。図3-14より次のような信頼度表現式を導くことができる。

$$F(G_c) = X_b X_e (X_c + \overline{X}_c X_f) (X_s + \overline{X}_s X_g)$$

$$\cdot X_j (X_h + \overline{X}_h X_k) (X_l + \overline{X}_l X_m)$$

$$+ X_b X_e (X_c + \overline{X}_c X_f) \overline{X}_s \overline{X}_g X_h$$

$$\cdot X_j (X_l + \overline{X}_l X_m)$$

$$+ \overline{X}_b X_c X_d X_e (X_s + \overline{X}_s X_g)$$

$$\cdot X_j (X_h + \overline{X}_h X_k) (X_l + \overline{X}_l X_m)$$

$$+ \overline{X}_b X_c X_d X_e \overline{X}_s \overline{X}_g X_h X_j X_j (X_l + \overline{X}_l X_m)$$

$$(3-37)$$

3. 3節で提案した木生成法を図3-10に適用すれば, 積項数 30 , 演算記号数 280 (リテラル数 281) の信頼度表現式が得られる. 一方, アサイクリックサブグラフ法によって得られる信頼度表現式は, 積項数 4 , 演算記号数 53 (リテラル数 54 ) となる.

したがって, アサイクリックサブグラフ生成法は木生成法と比べ,

- (i) 表現式を高速に生成できる,
- (ii) 簡単な表現式を導くことができる, という特徴を持っている.

### 3.5 積項数の最小化

本節では,グラフの強連結部分に注目してセクショングラフを作成するとき,グラフの信頼度表現式がセクショングラフの表現式の積として得られることを示す.

本節で提案するセクショングラフ法の概略を例題を用いて説明しよう. 式(3-37)を因数分解すれば次式が得られる.

$$F (G_c) = X_b X_e (X_c + \overline{X}_c X_f) (X_a + \overline{X}_a X_g)$$

$$\cdot X_j (X_h + \overline{X}_h X_h) (X_l + \overline{X}_l X_m)$$

$$+ X_b X_e (X_c + \overline{X}_c X_f) \overline{X}_a \overline{X}_g X_h$$

$$\cdot X_{i}X_{j} \left( X_{l} + \overline{X}_{l}X_{m} \right)$$

$$+ \overline{X}_{b}X_{c}X_{d}X_{e} \left( X_{a} + \overline{X}_{b}X_{g} \right)$$

$$\cdot X_{j} \left( X_{h} + \overline{X}_{h}X_{k} \right) \left( X_{l} + \overline{X}_{l}X_{m} \right)$$

$$+ \overline{X}_{b}X_{c}X_{d}X_{e}\overline{X}_{a}\overline{X}_{g}X_{h}X_{i}X_{j} \left( X_{l} + \overline{X}_{l}X_{m} \right)$$

$$= \left\{ X_{b}X_{e} \left( X_{c} + \overline{X}_{c}X_{f} \right) + \overline{X}_{b}X_{c}X_{d}X_{e} \right\}$$

$$\cdot \left\{ \left( X_{a} + \overline{X}_{a}X_{g} \right) X_{j} \left( X_{h} + \overline{X}_{h}X_{k} \right) + \overline{X}_{a}\overline{X}_{g}X_{h}X_{i}X_{j} \right\}$$

$$\cdot \left( X_{l} + \overline{X}_{l}X_{m} \right)$$

$$(3-38)$$

式(3-38)の演算記号数は 23 (リテラル数は 24) であり,式(3-37)のそれより大幅に削減できている.

図3-10のグラフG。の表現式の導出について考える.

グラフG。に対して,強連結成分を1つの頂点として得られる縮約グラフG。\*を図3-11に示す.縮約グラフG。\*はアサイクリックグラフであり,その頂点をそれぞれ終点とする枝集合が次のように示せる.

$$E_{t}(s) = \phi, E_{t}(v^{*}_{2}) = \{b, c\}$$
  
 $E_{t}(v^{*}_{3}) = \{a, g, h\}, E_{t}(v^{*}_{4}) = \{l, m\}$ 

式(3-32)を図3-11の縮約グラフに適用すれば、信頼度表現式 $F(G_c^*)$ が次のように直接導ける.

$$F (G_c^*) = \prod_{k=2}^{|V^*|} \sum_{e \in E_t(V^*_k)} X_e$$

$$= (X_b + X_c) (X_a + X_g + X_h) (X_l + X_m)$$

$$= (X_b + \overline{X}_b X_c) (X_s + \overline{X}_s X_g + \overline{X}_s \overline{X}_g X_h) (X_l + \overline{X}_l X_m)$$
(3-39)

式(3-39)は、3つの項が互いに素な関係にあるため、縮約グラフG。\*の信頼度は各枝の信頼確率を対応するブール変数に代入することによって求められる。

図3-10, 3-11から,  $v^*$ , (i=1, 2, 3, 4) に対応する G 。の頂点集合 V,が次のように示せる.

$$V_1 = \{s\}$$
 ,  $V_2 = \{v_2, v_3, v_4\}$   $V_3 = \{v_5, v_6, v_7\}$  ,  $V_4 = \{v_8\}$  すべての枝集合  $(V_j \times V_k) \cap E$  ,  $(j, k=1, 2, ..., i-1)$  を短絡

除去して得られる super source vertex s および頂点集合  $V_i$  から生成される セクショングラフを

$$G_i = \{ \{ s \} \cup V_i \}$$

と定める。なお、Vュに関するセクショングラフは

$$G_1 = \{ \{ s \} \} = \{ \{ s \} , \phi \}$$

と表される。図3-10に関するセクショングラフを図3-15に示す (super source vertex sを $\oplus$ で示す。以下同じ)。

図3-10および図3-15(a)おいて、ソースsから $V_2$ の任意の頂点に至る経路には枝b、cのうち少なくとも1本が含まれること、図3-10および図3-15(b)おいて、ソースsから $V_3$ の任意の頂点に至る経路には枝a, s, hのうち少なくとも1本が含まれること、また図3-10および図3-15(c)において、ソースsから $V_4$ の任意の頂点に至る経路には枝l、mのうち少なくとも1本が含まれることがいえる。また、式(3-39)が $G_c$ \*の頂点v\* $^2$ 2、v\* $^3$ 3、v\* $^4$ 4を終点とする並列枝の積として求められることから、 $G_c$ 0の信頼度表現式 $F(G_c)$ が次式で与えられる。

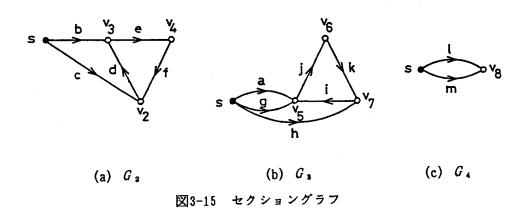
$$F(G_c) = \prod_{k=1}^{|V^*|} F(G_k)$$

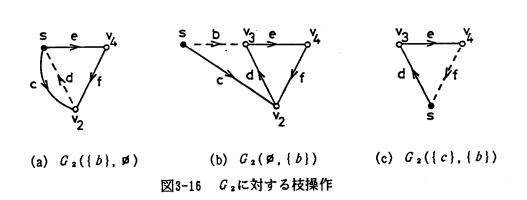
$$= F(G_2) \cdot F(G_3) \cdot F(G_4)$$
(3-40)

ここで、 $F(G_k)$  はセクショングラフ $G_k$ の信頼度表現式を示す.なお、 $F(G_1)=1$ である.

式(3-40)は、セクショングラフ単位に信頼度表現式を求めれば、それらの積から  $G_{\mathfrak{o}}$ の信頼度表現式が得られることを示している。セクショングラフ  $G_{\mathfrak{o}}$ の表現式は3.4節のアサイクリックサブグラフ法を適用すれば求めることができる。以下、簡単に説明する.

 $G_2$ において、ソース s を始点とする枝は  $b=(s,v_3)$ 、  $c=(s,v_2)$  の 2 個であることから、最初に短絡・開放操作可能な枝が b あるいは c に限られることは明かである。枝 b 、 c の短絡・開放操作について示そう。枝 b を選び、枝 b を短絡除去して得られるグラフを  $G_2$  ( $\{b\}$ ,  $\phi$ ) 、枝 b を開放除去して得られるグラフを  $G_2$  ( $\{b\}$ ) とおくとき、 $G_2$  の表現式 F ( $G_2$ ) が





次式のように書ける.

$$F (G_2) = x_b F (G_2 (\{b\}, \phi)) + \overline{x}_b F (G_2 (\phi, \{b\}))$$
(3-41)

 $G_2$ において,枝 b を短絡除去すれば,グラフ  $G_2$ ( $\{b\}$  , $\phi$ )が図3-16(a),枝 b を開放除去すれば,グラフ  $G_2$ ( $\phi$  ,  $\{b\}$  )が図3-16(b),のように得られる.なお,図3-16(b)における枝 b を点線で示しておく.図3-16(a)において,枝  $d=(v_2,s)$  のどのような状態も  $G_2$ ( $\{b\}$  , $\phi$ )の信頼度に影響することはなく,枝 d を  $G_2$  ( $\{b\}$  , $\phi$ ) から開放除去できる(開放除去した枝 d を点線で示す.以下同じ).  $G_2$  ( $\{b\}$  , $\phi$ ) から枝 d を開放除去すれば,頂点  $v_4$  , $v_2$  がサイクル上の頂点でなくなるため,得られるグラフはアサイクリックとなる.図3-16(a)に式(3-32)を適用すれば,F ( $G_2$  ( $\{b\}$  , $\phi$ ))が次のように得られる.

$$F(G_2(\{b\}, \phi)) = x_e(x_c + \bar{X}_c x_f)$$
 (3-42)

次に、枝 b を開放除去して得られるグラフ  $G_{2}$  ( $\phi$ ,  $\{b\}$ ) の信頼度表現式について考えよう(図3-16(b)参照). ソース s を始点とする枝は c=(s,s)

- $v_2$ )に限られるので,これを短絡除去すれば図3-16(c)に示す $G_2$ ( $\{c\}$ ,
- $\{b\}$ ) が得られる. 図3-16(c)において、枝fのどのような状態も $G_{2}$ ( $\{c\}$ ,
- $\{b\}$  ) の信頼度に影響することなく,枝 f を  $G_{2}$  ( $\{c\}$ ,  $\{b\}$ ) から開放除去できる。 $G_{2}$  ( $\{c\}$ ,  $\{b\}$ ) から枝 f を開放除去すれば,頂点  $v_{3}$ ,  $v_{4}$ がサイクル上の頂点でなくなるため,得られるグラフはアサイクリックとなる。図3-16(c)に式(3-32)を適用すれば,次式が得られる。

$$F(G_2(\{c\},\{b\})) = x_d x_e$$
 (3-43)

式(3-43)から $G_2$ ( $\phi$ , {b}) の表現式として、次式が求められる。

$$F (G_2 (\phi, \{b\})) = x_c x_d x_e$$
 (3-44)

したがって,式(3-42),(3-44)からG₂の信頼度表現式が次のように導ける.

$$F(G_2) = X_b X_e (X_c + \bar{X}_c X_f) + \bar{X}_b X_c X_d X_e$$
 (3-45)

また、 $G_{\bullet}$ 、 $G_{\bullet}$ の表現式も上述と同様な操作を繰り返せば、次のように得られる.

$$F (G_3) = (X_s + \bar{X}_s X_g) X_j (X_h + \bar{X}_h X_k) + \bar{X}_s \bar{X}_g X_h X_j X_j$$

$$(3-46)$$

$$F (G_4) = (X_l + \bar{X}_l X_m)$$

$$(3-47)$$

なお、枝を短絡・開放操作して得られたグラフがサイクルを含む場合、そのグラフを $G_c$ とおいて上述と同様な操作を繰り返せば信頼度度表現式が得られる。 グラフの信頼度表現式がセクショングラフの表現式の積(式(3-40)参照)と

して得られることから、積項数が1となる場合の条件を次に示す。

- (i) 縮約グラフの頂点数が3以上である場合. これは図3-10の場合に相当する.
- (ii) 縮約グラフの頂点数が2であり、与えられたグラフのsを始点とする枝数(並列枝も1と数える)が1である場合.

以上の議論から,図3-17に示すアルゴリズム SINGLE\_PRODUCT\_TERM を構築できる. 枝集合  $B_k$  および  $B_k$  の終点集合  $V_{k+1}$  (k=1, 2, …, c-1) に関して,  $|V_{k+1}| \neq 1$  はセクショングラフ  $G_{k+1}$  単位に表現式の導出が必要であることを,また  $|V_{k+1}| = 1$  は  $G_{k+1}$  の生成が不要であることを示している. 前者はセクショングラフ  $G_2$  ,  $G_3$  の場合に,また後者は  $G_4$  の場合に相当している.

図3-10にアルゴリズム SINGLE\_PRODUCT\_TERM を適用すれば,信頼度表現式として式(3-38)を導くことができる.

したがって、3.5節で提案したセクショングラフ法は

- (i) 信頼度表現式がセクショングラフの表現式の積として導ける,
- (ii) そのため3.4節で提案した算法よりコンパクトな(因数分解された,リテラル数が少ない)信頼度表現式が得られる,

という特徴を持っている.

# 3.6 実験結果と考察

以下に示す実験結果はそれぞれのアルゴリズムを FORTRAN でプログラムし、 愛媛大学情報処理センター FACOM M-360AP を使用して行ったものである. ここ で示す演算時間および積項数は、指定した枝数に対してそれぞれ 10 個のラン

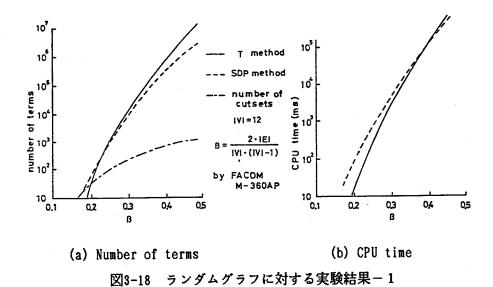
```
procedure SINGLE PRODUCT_TERM(G=(V,E),s)
    begin
        procedure GROW(G(E^+,E^-),G[V],l,F_l)
            comment: G[V] is a vertex-section subgraph of G(E^+,E^-);
            begin
                 delete arcs directed toward a super source vertex s;
                 construct the condensed graph G^*(V^*);
                 find a topological order a;
                 determine the partitions of V and N pertinent to \alpha:
                      \{V_1, V_2, \dots, V_c\}, \{N_1, N_2, \dots, N_c\};
                                     B_k \triangleq (\sum_{j=1}^{n} V_j, V_{k+1}), k=1,2,\dots,c-1;
                 form arc sets
                 comment: if e \in B_k is a multiple arc consisting of
                                  e_1,e_2,e_3,\cdots, we write E \cup \{e\} = E \cup \{e_1,e_2,e_3,\cdots\},
                                  E^+ \cup \{e\} = E^+ \cup \{e_1 \parallel e_2 \parallel e_3 \parallel \cdots \}, x_e = x_{e_1} x_{e_2} x_{e_3} \cdots,
                                  x_e = x_{e_1} + x_{e_1}x_{e_2} + x_{e_1}x_{e_2}x_{e_3} + \cdots;
                 for k:=1 until c-1 do
                     begin
                         if |V_{k+1}|=1 then F_r:=x_e, where B_k=\{e\};
                         else
                              begin
                                  F_{\tau}:=0;
                                  E^-:=\emptyset;
                                  X:=1;
                                  for i:=1 until |B_k| do
                                      begin
                                          GROW(G(E^{+} \cup \{e_i\}, E^{-}), G[V_{k+1}], l+1, 1)
                                                   where e_i=(s,v) \in B_k, v \in V_{k+1};
                                          F_{\tau} := F_{\tau} + X * x_{e_i} * F_{l+1};
                                           E^{-}:=E^{-}\cup\{e_i\};
                                           X:=X*\overline{x}_{e}
                                      end
                              end;
                          E^+:=E^+\cup B_k\cup ((V_{k+1}\times V_{k+1})\cap E);
                         F_l := F_l * F_\tau
                     end
             end GROW:
        GROW(G(\phi,\phi),G(V,E),1,1);
        output formula F_1
    end SINGLE_PRODUCT_TERM;
```

図3.17 アルゴリズム SINGLE\_PRODUCT\_TERM

ダムグラフを発生し、その結果を平均したものである.

## 3.6.1 木生成法

- 3.3節で提案した木生成法(tree generation method, 簡単のため T method)と,極小枝集合としてカットセットを用いたSDP法の比較実験の結果を示す。図3-18は頂点数12の無向グラフ,図3-19はp=1.3, 1.5(p=|E|/|V|)の無向グラフに関する実験結果である。
  - この結果から、次のことがいえる.
  - (i) 与えられたグラフにおいて、カットセットの個数は木の個数に比べ 圧倒的に少ないのが一般的である(例えば、頂点数 |V| の完全グラフではカットセットの個数が $2^{|V|-1}$ であるのに対し、木の個数は  $|V|^{|V|-2}$ にもなる)。カットセットを用いたSDP法は式(3-7)の式(3-9)による展開によって多数の積項が生成されるため、その積項数は指数的に増加し、木の個数に匹敵することになる。木生成法によって得られる積項数は与えられたグラフの木の個数と等しくなるが、一般のグラフではカットセットの個数が木の個数よりも少ないため、カットセットを用いたSDP法により得られる積項数が木の個数よりも少なくなることが多い。 $\beta$ あるいはpの値が小さな場合を除けば、図3-18(a)および図3-19(a) (p=1.5) はそのことを示している。しかしながら、SDP法による積項数はグラフの形状から一意に定まらず、たとえば図3-20の ARPA ネットワーク( $\beta=0.43$ )のように、木生成法による積項数(木の個数)がSDP法による積項数よりも少なくなる場合もある(表3-2参照)。
  - (ii) βあるいはρの値が小さい場合,木生成法はSDP法より高速にSAT信頼度表現式を導出できる。それは、SDP法ではブール積の比較演算が必要であり、これに多大な演算時間を要するのに対し、木生成法では比較演算が不要であり、木の生成と同時に積項が直接算出できるためである。



10 105 10 \_≜ SDP method 10 number of terms number of CPU time (ms) by FACOM M-360AP ري 10 10 10 18 10 18 14 IV! 16 12 (b) CPU time (a) Number of terms 図3-19 ランダムグラフに対する実験結果-2

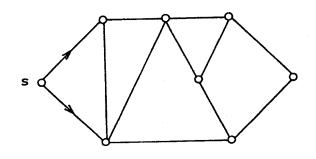


図3-20 ARPA ネットワーク

表3-2 ARPA ネットワークの実験結果

	SDP法	木生成法
積項数	296	247
CPU時間(msec)	154	24

また,表現式算出に必要なメモリに関して次のことがいえる.

SDP法は、カットセットに対応するブール積項を列挙したのち、それらを互いに素な積項の和に展開する過程で生成される中間的積項をすべて記憶する必要がある。それに比べ、木生成法は生成木上においてその root から処理中の頂点に至るパス上の頂点に関する G ( $E^+$ ,  $E^-$ ) のみを記憶すればよい。

### 3.6.2 アサイクリックサブグラフ生成法

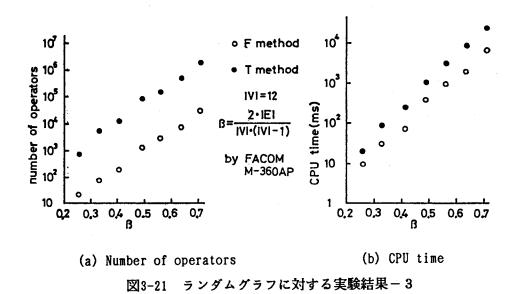
3. 4節で提案したアサイクリックサブグラフ生成法(factored formula method, 簡単のため F method)と、3. 3節で提案した木生成法(tree generation method, 簡単のため T method)の比較実験結果を以下に示す. 図 3-21は頂点数12の有向グラフに関する実験結果である.

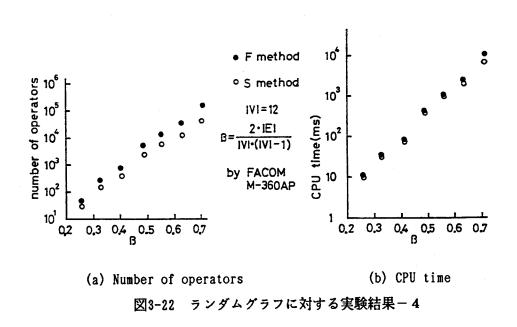
その結果から、次のことがいえる、

- (i) 木生成法によって得られる表現式の積項数は木の個数と等しくなるので、その演算記号(リテラル)数は少なくとも頂点数と木の個数の積となる。一方、アサイクリックサブグラフ生成法によって得られる表現式の積項数が枝の短絡・開放操作により生成されるアサイクリックサブグラフ数に等しいこと、および得られる表現式が factored form の形式で得られるため木生成法に比べ大幅に演算記号(リテラル)数を減少できる。
- (ii) 木生成法では木をすべて生成するのに対して、アサイクリックサブ グラフ生成法では頂点に対する位相的順序を用いることにより不要な 操作を削減できるため、演算時間を減少できる.

### 3.6.3 セクショングラフ法

3.5節で提案したセクショングラフ法 (single product term method, 簡単のため S method) と, 3.4節で提案したアサイクリックサブグラフ生成法 (factored formula method, 簡単のため F method) の比較実験結果を以下





に示す. 図3-22は頂点数12の有向グラフに関する実験結果である. この結果から、次のことがいえる.

- (i) アサイクリックサブグラフ生成法によって得られる表現式の積項数は枝の短絡・開放操作により生成されるアサイクリックサブグラフ数と等しくなるので、その演算記号(リテラル)数は少なくとも頂点数とアサイクリックサブグラフ数の積となる。一方、セクショングラフ法によって得られる表現式はセクショングラフの表現式の積として求められるため、因数分解されたコンパクトな形式となる。このため、アサイクリックサブグラフ生成法に比べ演算記号(リテラル)数を大幅に減少できる。
- (ii) アサイクリックサブグラフ生成法では枝の短絡・開放操作によりアサイクリックサブグラフをすべて生成するのに対し、セクショングラフ法ではセクショングラフ単位に枝の短絡・開放操作を行うため、枝に対するの重複操作を避けることができ、演算時間を減少させることができる。

### 3.7 結言

本章では、ネットワークのSAT信頼度を求めるアルゴリズムについて議論 し、3つの算法を提案した。

(1) 3.3節では、ネットワーク構造を持つシステムを正常(異常)とするようなシステム成分の極小集合から、システムの信頼度(非信頼度)表現式を導出する算法(SDP法)について概説し、さらに得られる表現式の良さを定める目的関数を積項数におくとき、最適な表現式を得るために極小集合が具備すべき条件およびその処理順序について考察し、特に極小集合の個数に等しい積項数を持つ表現式算出のための条件を示した。また、枝の短絡・開放操作を基本とした木の生成法を述べ、これらの木( $E^+$ )の生成とそれに付随して得られる枝の集合( $E^-$ )の情報から、ブール演算を行うことなく、木の総数に等しい積項数を持つSAT信頼度表現式を導く算法を示した。木生成法は積項

数が木の個数に制限される欠点を持つが、SDP法のようなブール演算が不要であるため、演算に必要なメモリが少なく、積項当りの算出時間が高速である、という特徴を持つ、

- (2) サイクルを含まない有向グラフの頂点に対して位相的順序で整数が割り当てられること、この順序にしたがい枝(あるいは並列枝)を短絡操作して得られる信頼度表現式の積項数がグラフの規模に関わりなく1となることを示した。一般のサイクルを含む有向グラフに関して、3.4節では枝に対する短絡・開放操作からアサイクリックサブグラフを生成し、信頼度表現式がサブグラフに対応する積項の和として得られることを示した。提案した算法は、得られる表現式の積項数が枝の短絡・開放操作により生成されるアサイクリックサブグラフの個数に等しいこと、および得られる表現式がfactored form の形式で得られるため木生成法に比べ大幅に演算記号数を減少できる、という特徴を持つ。また、アサイクリックサブグラフ生成法では頂点に対する位相的順序を用いることにより枝に対する操作回数が削減でき、大幅に演算時間を減少させることができた。
- (3) 3.5節では、与えられたグラフをセクショングラフに分割するとき、グラフの信頼度表現式がセクショングラフの表現式の積として得られること、さらに積項数1の表現式が得られるグラフの条件、を示した。また、セクショングラフ法によって得られる信頼度表現式が因数分解されたコンパクトな形式となるため、アサイクリックサブグラフ生成法に比べ表現式に含まれる演算記号数を大きく削減できる、ことを示した。さらに、3.5節で提案した算法はセクショングラフ単位に表現式を導けばよいため、枝に対する重複操作を避けることができ、演算時間を減少させることができた。

以上の考察から、SAT信頼度表現式を求める算法のうち、セクショングラフ法が最も優れていることが判明した。

# 第4章 **k**−out-of-**n**システム信頼度

#### 4.1 緒言

与えられたシステムにおいて、システムに含まれる $\pi$ 個の要素のうち、少なくとも $\pi$ 6 個の要素が正常(異常)ならば、システムが正常(異常)となるものを、 $\pi$ 6 の u t - o f -  $\pi$ 1 :  $\pi$ 6 ( $\pi$ 7) システムという。このようなシステムの例として、少なくとも $\pi$ 6 個の入力が"1"であれば、出力が"1"となる論理回路が考えられ、特に $\pi$ 6 1 の場合この論理回路は $\pi$ 7 Rゲートであり、 $\pi$ 8 の場合は $\pi$ 7 N D ゲートである[46]。

要素のそれぞれに信頼確率が与えられ、さらにある要素が異常となっても他の要素の信頼確率に影響を与えることがないという制約(統計的独立という)のもとで、システムが正常(異常)である確率-システムの信頼度(非信頼度)-を、相異なる k 個の正常(異常)な要素だけからなる組が少なくとも1つ存在する確率として求めることができる。

4. 2節では本章で使用する用語,記法を定める. 4. 3節では k-out -of-nシステムの信頼度表現式導出に木展開法の応用を試みる[48,49,51,52]. 4. 4節では近似信頼度を求めるアルゴリズムを提案する[54,55,56].

### 4.2 諸定義

ここでは、4.3節以降で使用する記号および用語等の説明を行う.

 $n: \lambda > \lambda > \lambda$  内の要素数、要素には $1 h \in \Lambda$  までの番号を付与する、

k:システムを正常(異常)とする最小の要素数.

l: n-k+1.

π: k個の要素を番号の小さなものから順に並べた順序列。

M:順序列πの総数で, ( %) 個ある\*.

 $X_i$  ( $\bar{X}_i$ ):要素iが正常(異常)であることを示すブール変数.

 $P_i(q_i)$ :要素iが正常(異常)である確率.

 $A:\pi$  に属する要素 i に対応するブール変数  $x_i$ の積.

Pr {A}: Aの信頼度.

R (U):システムの信頼度(非信頼度).

 $B_R(B_v)$ : システムの信頼度(非信頼度)のブール表現.

[定義4-1 ] [1,46,47] システムに含まれる $\pi$  個の要素のうちk 個の要素が正常 (異常) ならば、システムが正常 (異常) となるものを、k-out-of-n:G(F) システムといい、 $\pi G_k(\pi F_k)$  と書く.

[定義4-2] [1,47]  $\pi$  に属する k 個の要素が同時に正常(異常)であるとき,k-o u t-o f $-\pi$ システムは正常(異常)となるので, $\pi$  を極小な順序列(Gシステムでは極小パス,Fシステムでは極小カットに相当)と呼ぶ。これら M 個の $\pi$  は小さな要素番号を含むものから優先して順番を付けるものとする。

定義4-2のように極小な順序列を定めれば、k-0 u t -0 f  $-\pi$  システムの極小な順序列は根を持つ1つの有向木 Tで表現できる。

[例題4-1] 3-out-of-5システムにおいて、5個の要素を1,

2, 3, 4, 5 とおけば, ( $\frac{5}{3}$ ) = 10 個の極小な順序列は,

 $\pi_1 = \{1, 2, 3\}$ 

 $\pi_2 = \{1, 2, 4\}$ 

 $\pi_3 = \{1, 2, 5\}$ 

 $\pi_4 = \{1, 3, 4\}$ 

 $\pi_{5} = \{1, 3, 5\}$ 

 $\pi_6 = \{1, 4, 5\}$ 

 $\pi_7 = \{2, 3, 4\}$ 

<sup>\*</sup>  $\binom{n}{k} = n! / \{ (n-k)! \cdot k! \}$ 

$$\pi_8 = \{2, 3, 5\}$$

 $\pi_9 = \{2, 4, 5\}$ 

 $\pi_{10} = \{3, 4, 5\}$ 

の順序で表せるので、これらは図4-1に示す有向木 Tで表現できる。

以下では、有向木における子、兄弟、葉、根、深さ等の用語は従来の慣例に 従うものとする[50]. ここで兄弟のうち右側の頂点を左側のそれの兄と定める.

[定理 4-1 ]  $\begin{bmatrix}1,9,47\end{bmatrix}$   ${}_{n}G_{k}$   $\left({}_{n}F_{k}\right)$  システムは  ${}_{n}F_{l}$   $\left({}_{n}G_{l}\right)$  システムと 等価な システムである.

[定義4-3]  $_nF_k$ システムにおいて,n個の要素が相異なる信頼確率を持つシステムの信頼度をR ( $p_1 \sim p_n$ ; k) と書き,またすべての要素が同一の信頼確率pを持つシステムの信頼度をR (n, p, k) と書くことにする.

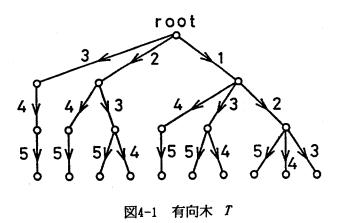
## 4.3 木展開法の応用

k - out - of - nシステムの信頼度(非信頼度)を求める算法として,

- (i) 直接展開法[1],
- (ii) 改良IE法 (improved method of inclusion-exclusion) [47],
- (iii) 繰返しIE法 (inclusion-exclusion method) [9],
- (iv) SDP法[15] (sum of disjoint products method), がある.
- (i) は,すべての k 個の要素の組 $\pi$ 1,  $\pi$ 2, …, $\pi$  kを列挙し,2 k-1 個の組み合せすべてを考えるものであり,その組み合せには互いに相殺可能な積項を多く含むことになる[48,49]. (ii) は (i) における互いに相殺されない積項だけを求めるものであるが,得られるブール表現には積項の論理和および論理減算を含むことになる.一般のネットワーク構造を持つシステムに関して,(iii),(iv) が木展開法に比べて劣ることは第2章で既に報告している.

本節では、システムの信頼度(非信頼度)表現式の算出に関して、改良 I E 法に比べ、

(i) 得られる積項数が少なく、特に指定したnに対してkがn/2近く



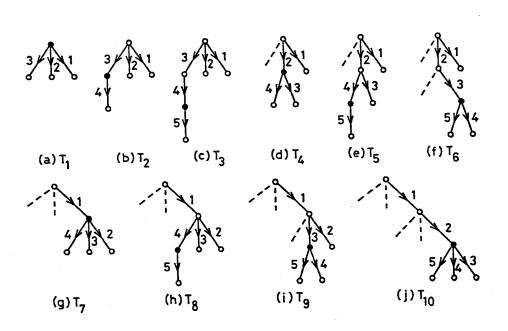


図4-2 積項木

になるほどその差が顕著になる.

(ii) 表現式が積項の和形式で得られる,

という特徴を持つグラフ理論的手法(木展開法)[49,51,52]を提案する.

### 4.3.1 既存の定理と本手法の関係

ここでは,第2章で提案している木展開法と4.3節で提案する木展開法の 関係を示す.

2 頂点 s, t を持つ信頼性グラフにおいて,すべての s - t パスを求め,これを  $\mu_i$  (i = 1, 2, …,  $\mu$ ) と名づける。  $\mu_i$ 上に始点および終点を持ち,  $\mu_i$ と互いに素な枝から構成される  $\mu_i$  (j = 1, 2, …, i - 1) の部分パス (付加パスという) を  $\mu_i$  とおく (詳細は 2 章の 2 3 ページ参照)。  $\mu_i$  および  $\mu_i$  に対応するブール変数の積をそれぞれ  $A_i$  、  $A_i$  とおけば,その s - t 間の信頼度表現式  $B_s$  は次式で表せる。

$$B_R = A_1 + A_2 + \cdots + A_m \tag{4-1}$$

また, 信頼性グラフの信頼度 R は次式のように導ける.

$$R = P \left( \mathcal{G}_1 \right) + P \left( \mathcal{G}_2 \right) + \cdots + P \left( \mathcal{G}_m \right) \tag{4-2}$$

ここで、式(4-2)の  $\mathcal{G}_i$ は  $\mu_i$ および  $\mu_{Ji}$  (j=1, 2, …, i-1) で作られる s-t アサイクリックグラフ[4,11]の集合であり、P ( $\mathcal{G}_i$ ) は次式で与えられる.

$$P (\mathcal{G}_i) = P r \{A_i\} (1 - P r \{A_{1i} + A_{2i} + \cdots + A_{i-1, i}\})$$

$$(4-3)$$

 $P(\mathcal{G}_i)$  のブール表現を $B_i$ で表せば、式(4-3)は次のように示せる.

$$B_{i} = A_{i} \left( \overline{A_{1i} + A_{2i} + \cdots + A_{i-1, i}} \right)$$

$$= \overline{A}_{1i} \overline{A}_{2i} \cdots \overline{A}_{i-1, i} A_{i}$$
(4-4)

付加パスの集合 $\mu_{ji}$  (j=1, 2, …, i-1) を, ブール表現  $T_i$ により,

$$T_{i} = A_{1i} + A_{2i} + \cdots + A_{i-1,i}$$
 (4-5)

と表せば、式(4-4)は次のように書ける.

$$B_i = \overline{T}_i A_i \tag{4-6}$$

式(4-6)の  $T_i$ を展開して得られる各積項は,有向木  $T_i$ の部分木(積項木という)と 1 対応している.この性質を利用し,第 2 章では積項木の生成から直接  $T_i$ の各積項を算出している.これを木展開法という.

一方, 信頼性グラフのs-t間の信頼度Rは,

$$R = P r \{ A_1 + A_2 + \cdots + A_m \}$$
 (4-7)

と表せるので、非信頼度 Uは次のように書ける.

$$U = 1 - R$$
= 1 - P r { A<sub>1</sub> + A<sub>2</sub> + ··· + A<sub>m</sub>} (4-8)

パス集合 $\mu_i$  (i=1, 2, …,  $\mu$ ) から生成される有向木 I に関するブール表現を、

$$T = A_1 + A_2 + \cdots + A_m \tag{4-9}$$

と表すことにすれば、式(4-6)と同様に非信頼度表現式 $B_v$ は次のように書ける.

$$B_{U} = (\overline{A_{1} + A_{2} + \cdots + A_{m}})$$

$$= \overline{T}$$
(4-10)

式(4-10)は式(4-6)の  $\overline{T}_i$ の展開と同様,有向木 Tに木展開法を適用すれば, $\overline{T}$ の各積項が Tの積項木から直接算出できることを示している.

以上から、k-out-of-nシステムに関しても、同様な式を導くことができる。

[定理 4-2 ]  $_{n}G_{k}$ システムの非信頼度表現式  $B_{v}(G)$  は,有向木 Tに関する表現を,

$$T(G) = A_1 + A_2 + \cdots + A_M$$
 (4-11)

とおくとき、次式で表せる.

$$B_{v} = \overline{T} (G) \tag{4-12}$$

また、 $_nF_k$ システムの信頼度表現式  $B_R$  (F) は、有向木 T に関するブール表現を変数の補元 (complement) の積和 T (F) とおくとき、次式で表せる.

$$B_R(F) = \overline{T}(F) \tag{4-13}$$

図4-1に関して、 $_{5}G_{3}$ システムのT(G)は、

$$T (G) = [x_1 \{x_2 (x_3 + x_4 + x_5) + x_3 (x_4 + x_5) + x_4 x_5\}]$$

$$+ [x_2 \{x_3 (x_4 + x_5) + x_4 x_5\}]$$

$$+ x_3 x_4 x_5$$

$$(4-14)$$

と表せるので、 $B_{\sigma}(G)$  は次のように書ける.

$$B_{V}(G) = \{ \overline{X}_{1} + X_{1} (\overline{X}_{2} + X_{2} \overline{X}_{3} \overline{X}_{4} \overline{X}_{5}) \\ \cdot (\overline{X}_{3} + X_{3} \overline{X}_{4} \overline{X}_{5}) (\overline{X}_{4} + X_{4} \overline{X}_{5}) \} \\ \cdot \{ \overline{X}_{2} + X_{2} (\overline{X}_{3} + X_{3} \overline{X}_{4} \overline{X}_{5}) (\overline{X}_{4} + X_{4} \overline{X}_{5}) \} \\ \cdot (\overline{X}_{3} + X_{3} \overline{X}_{4} + X_{3} X_{4} \overline{X}_{5})$$
(4-15)

これを展開すると,次式が得られる.

$$B_{U}(G) = \overline{X}_{1} \, \overline{X}_{2} \, \overline{X}_{3} + \overline{X}_{1} \, \overline{X}_{2} \, X_{3} \, \overline{X}_{4} + \overline{X}_{1} \, \overline{X}_{2} \, X_{3} \, X_{4} \, \overline{X}_{5} + \overline{X}_{1} \, X_{2} \, \overline{X}_{3} \, \overline{X}_{4}$$

$$+ \overline{X}_{1} \, X_{2} \, \overline{X}_{3} \, X_{4} \, \overline{X}_{5} + \overline{X}_{1} \, X_{2} \, X_{3} \, \overline{X}_{4} \, \overline{X}_{5} + X_{1} \, \overline{X}_{2} \, \overline{X}_{3} \, \overline{X}_{4}$$

$$+ X_{1} \, \overline{X}_{2} \, \overline{X}_{3} \, X_{4} \, \overline{X}_{5} + X_{1} \, \overline{X}_{2} \, X_{3} \, \overline{X}_{4} \, \overline{X}_{5} + X_{1} \, X_{2} \, \overline{X}_{3} \, \overline{X}_{4} \, \overline{X}_{5}$$

$$(4-16)$$

また、 $_{5}F_{3}$ システムの $B_{R}(F)$  は、式(4-16)のブール変数x,を $\overline{x}$ ,に、 $\overline{x}$ , をx,に変換すれば得られる.

図4-2の  $T_1 \sim T_{10}$ は,式(4-16)の各積項を有向木 Tの部分木として表現したもので,それぞれを積項木と呼ぶ.ここで,式(4-16)の各積項は,対応する積項木の葉に接続される枝ラベル I に関して,そのブール変数の補元( $\overline{X_i}$ )をとっている.

図4-2の点線は、式(4-16)の展開過程において生じる積項あるいは変数の抹消に対応して、木の枝および終点を開放除去すること(欠落)を示している.

欠落を考慮しながら,有向木 Tから積項木を連続的に生成するグラフ理論的 方法を示す.

本システムにおける有向木 Tは次の性質を持っている.

(i) T上の葉でない頂点vにおいて、vに入る枝のラベルおよびvから 出る枝のラベルの集合を、それぞれ $e_{in}(v)$ 、 $E_{out}(v)$  とおけば、 $v_i$ とその弟 $v_{i+1}$ の間には次の関係が成立する。

$$E_{out}(v_i) = E_{out}(v_{i+1}) + e_{in}(v_{i+1})$$
 (4-17)

(ii)  $v_i$ の弟の集合を  $V_v(v_i)$  , 兄の集合を  $V_e(v_i)$  , および一番若

い弟を w,とおけば,式(4-17)より次式が成立する.

$$E_{out}(v_i) = \sum_{w \in V_y(v_i)} e_{in}(w) + E_{out}(w_y) (4-18)*$$

(iii) Tにおいて、根から葉に至るパス上のすべての頂点に関して、その 兄を終点とする枝のラベルおよびパスを構成する枝のラベルのそれぞ れは互いに相異なる.

Tの部分グラフT。を次のように定義する.

- (i) Tの根を含み連結である。
- (ii)  $T_s$ のどの頂点も、T上における兄弟をすべて含む、

Tの性質から明らかなように,T。は一般に同じラベルの枝を持ち得る.もし,T。の枝のラベルがすべて異なれば,それは1つの積項木となることは明かである[8, 20, 21].以下では,重複ラベルを持つT。を積項木T,に修正することについて述べる.

Tの部分木 T。において,根を始点とし葉を終点とするパス(ただし,葉に兄弟があるときは,一番若い葉を終点とする)を T。上の指定したパスと定める(図4-3参照,太線は指定したパスを示す).パス上の終点でない任意の頂点をVとおけば, T。は次の性質を持つ.

(1)  $u \in V_e(v)$  において、 $E_{out}(u) \neq \phi$ ならば性質(i)より、 $E_{out}(u) = E_{out}(v) + \sum_{z \in V} e_{in}(z) + e_{in}(v)$ 

(4-19)

が成立するので、 $I_s$ からuを根とする極大部分グラフが除去できる、ただし、 $V=V_s$  (v)  $\cap V_v$  (u) とする.

(2) 性質(ii)より,次式が成立する.

$$E_{out}(v) \underset{w \in V_{v}(v)}{\supset} \Sigma \qquad e_{in}(w)$$
 (4-20)

さらに、パス上のvの子から終点に至るすべての頂点について、それ自身およびその兄を終点とする枝ラベルの和集合 $E_{\alpha}(v)$ は次の関係を持つ。

<sup>\*</sup> Σは直和を意味する.

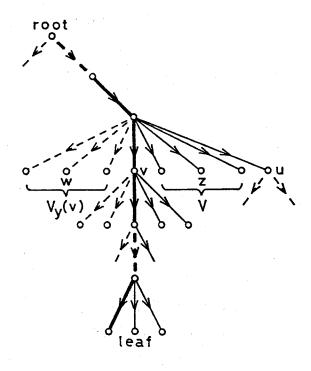


図4-3 Tの部分木 T。

vは任意であるから、パス上の葉を除くすべての頂点で式(4-21)が成立する. したがって、すべての頂点vに対して、 $e_{in}$ (w)、 $w \in V_v$ (v) が除去できる. $T_s$ の性質(1)、(2)、およびTの性質(iii) より、 $T_s$ から除去されずに残った枝で構成される木 $T_v$ (図4-3の指定したパスおよび実線で描かれた枝から成る木)は、異なるラベルを持つことができる.例えば、図4-2(f)の $T_s$ はパス(2、3、4)に関して、上記 $T_v$ の性質を満たしている.

以上より、積項木の生成手続きを次のように示せる.

- (1) Tの根とその子だけから成る積項木 $T_1$ とおく.
- (2) T<sub>i</sub>から積項木 T<sub>i+1</sub>の生成
  - (i)  $T_i$ 上において、左優先で葉vを選ぶ。
  - (ii) T上でvの子が存在する場合  $T_i$ 上のvに対して,T上におるvの子を付け $T_{i+1}$ を作る.これを vの探索(traverse)という.i:=i+1とおいて(i)へ.
  - (iii) T上でvの子が存在しない場合
    - (a)  $T_i$ から、vおよびvを終点とする枝を除去したものを $T_i$  とおき、vの親をuとおく。
    - (b)  $u \in V$ 以外の子があれば(i)へ.
    - (c) uにv以外の子がない場合 uが根ならば操作は終了し, uが根でなければv := uとおいて(a)  $\sim$ .

以上の手続きによって生成された積項木に対応する積項は、G(F)システムの場合、積項木を構成している枝ラベルiに対応するブール変数 $x_i(\bar{x_i})$ の積として求められる。ただし、葉を終点とする枝ラベルiに関しては $\bar{x_i}(x_i)$ をとるものとする。

図4-1の Tから図4-2の  $T_1$ ~  $T_{10}$ が生成される過程を表4-1に示す.図4-2における $\oplus$ 印の頂点は探索した頂点を示す.

 $T_i$ の枝集合として格納される枝の順序は,Tを深さ優先探索(depth-first search)により右側優先で探索したときの枝の探索順序に矛盾しないものとす

表4-1 Tの積項木

i	T;の枝集合	積項木	積項
1	1,2,3	T 1	$\overline{X}_1$ $\overline{X}_2$ $\overline{X}_3$
2	1,2,3,4	T 2	$\overline{X}_1 \ \overline{X}_2 \ X_3 \ \overline{X}_4$
3	1,2,3,4,5	Т 3	$\overline{X}_1 \overline{X}_2 X_3 X_4 \overline{X}_5$
	1,2,3,4		
	1,2,3		
	1,2		
4	1,2,3,4	T 4	$\overline{X}_1 X_2 \overline{X}_3 \overline{X}_4$
5	1,2,3,4,5	Т 5	$\overline{X}_1 X_2 \overline{X}_3 X_4 \overline{X}_5$
	1,2,3,4		
	1,2,3	£	
6	1,2,3,4,5	Т в	$\overline{X}_1 X_2 X_3 \overline{X}_4 \overline{X}_5$
	1,2,3,4	٠	
	1,2,3		
	1,2		
	1		
7	1,2,3,4	Т,	$X_1 \overline{X}_2 \overline{X}_3 \overline{X}_4$
8	1,2,3,4,5	T 8	$X_1 \overline{X}_2 \overline{X}_3 X_4 \overline{X}_5$
	1,2,3,4		
	1,2,3		
9	1,2,3,4,5	<i>T</i> 9	$X_1 \overline{X}_2 X_3 \overline{X}_4 \overline{X}_5$
	1,2,3,4		
	1,2,3		:
	1,2		
1 0	1,2,3,4,5	T 10	$X_1 X_2 \overline{X}_3 \overline{X}_4 \overline{X}_5$

る. このようにすると、少ない探索で次の積項木を生成することができる. 以上から、スタック上の操作だけで、 $B_{\sigma}(G)$ を互いに素な積項に展開するアルゴリズムを図4-4のように示せる.

# 4.3.2 gGkシステムのBvの一般形

図4-1の有向木 Tの頂点を探索することにより,図4-2の積項木が生成される.  $G_3$ システムの非信頼度表現式  $B_v$ は,これら積項木に 1 対 1 対応する積項の和として,式(4-16)のように得られる.また,この式の各積項を要素数で分類すれば表4-2が得られる.したがって, Tの根からある頂点までの距離をその頂点の深さとおけば,図4-1において,

- (i) 要素数3の積項は根(深さ0の頂点),
- (ii) 要素数4の積項は深さ1の頂点,
- (iii) 要素数5の積項は深さ2の頂点,

をそれぞれ探索することによって生成される.

ある頂点を探索すると、その頂点の弟とその子孫および先祖の弟とその子孫 が木から除去されるので、指定された要素数を持つ積項の総数は当該深さの頂 点数に等しくなる。例えば、要素数5の積項数および深さ2の頂点数はともに 6である。

また、要素数5の積項において、 $\bar{X}_5$ は共通であり、各積項とも $\bar{X}_5$ を除く要素のうち、深さと同数の補元でない要素がある。したがって、要素数が5である積項の数は( $\frac{4}{2}$ )で与えられる。

以上の考察から、 $_nG_n$ システムにおける積項数、要素数、および有向木 $_nG_n$ での関係が表4-3のように示せる。

表4-3から、すべての要素が等しい信頼確率 P (= 1 - q) を持つならば、非信頼度 U は次のように導かれる.

$$U = q^{-l} \sum_{i=0}^{k-1} {l+i-1 \choose i} p^{-i}$$
 (4-22)

```
procedure TEXT(n,k)
   begin
       l := n-k+1;
       for i:=1 until l do
          begin
              Term(i) := \overline{x}_i;
              Depth(i):=1;
              Son(i) := l - i + 1;
              output Term
          end:
       while i>0 do
          begin
              if Term(i) is a complemented variable, say \overline{x}_j,
                     and Son(i) \neq 0 then
                 begin
                     Term(i) := x_i;
                     for m:=1 until Son(i) do
                        begin
                            Term(i+m):=\overline{x}_{j+m};
                            Depth(i+m):=Depth(i)+1;
                            if Depth(i) < k-1
                                then Son(i+m):=Son(i)-m+1;
                                else Son(i+m):=0
                        end;
                     i:=i+Son(i);
                     output Term
                 end;
              else i:=i-1
          end
   end TEXT;
```

図4-4 アルゴリズム TEXT

表4-2 積項の分類

要素数	3	4	5
積項	$\overline{X}_1$ $\overline{X}_2$ $\overline{X}_3$	$ \overline{X}_{1} \ \overline{X}_{2} \ X_{3} \ \overline{X}_{4} $ $ \overline{X}_{1} \ X_{2} \ \overline{X}_{3} \ \overline{X}_{4} $ $ X_{1} \ \overline{X}_{2} \ \overline{X}_{3} \ \overline{X}_{4} $	$ \overline{X}_{1} \ \overline{X}_{2} \ X_{3} \ X_{4} \ \overline{X}_{5}  \overline{X}_{1} \ X_{2} \ \overline{X}_{3} \ X_{4} \ \overline{X}_{5}  \overline{X}_{1} \ X_{2} \ X_{3} \ \overline{X}_{4} \ \overline{X}_{5}  X_{1} \ \overline{X}_{2} \ \overline{X}_{3} \ X_{4} \ \overline{X}_{5}  X_{1} \ \overline{X}_{2} \ X_{3} \ \overline{X}_{4} \ \overline{X}_{5}  X_{1} \ \overline{X}_{2} \ X_{3} \ \overline{X}_{4} \ \overline{X}_{5} $

表4-3 要素数と積項数の関係

深さ	要素数	積項数
0	l	1
1	l+1	( l )
2	l+2	$\binom{l+1}{2}$
:	:	:
k-1	л	$\left(\begin{array}{c} n-1\\ k-1 \end{array}\right)$

いま、 $I_{I}$ 、 $X_{I}$ を次のように定める.

$$I_{j} \triangleq \{1, 2, \cdots, j\}$$
 (4-23)

$$X_{j} \triangleq \{X_{i} \mid i \in I_{j}\} \tag{4-24}$$

[定義4-4] B(j,r)は $\binom{i}{r}$ )個の互いに素な積項の和であり,各積項はj+1個の相異なる変数を含むものとする。その変数の内訳は,1個の $\overline{x}_{j+1}$ , $X_j$ に含まれる変数のうちr個の相異なる $x_i$ ,および $X_j$ の残り(j-r個)の変数 $x_i$ の補元 $\overline{x_i}$ である。

定義4-4,表4-3から, $_nG_k$ システムの非信頼度表現式 $B_u$ (G)が次式で与えられる.

$$B_{U}(G) = \overline{X}_{1} \overline{X}_{2} \cdots \overline{X}_{l} + \sum_{i=1}^{k-1} B(l+i-1, i)$$
 (4-25)

[例題4-2]  $_5G_3$ システムのB(・),  $B_v$ (G) は次のように導ける. ただし, l=3 である.

$$B (3, 1) = \overline{X}_{1} \, \overline{X}_{2} \, X_{3} \, \overline{X}_{4} + \overline{X}_{1} \, X_{2} \, \overline{X}_{3} \, \overline{X}_{4} + X_{1} \, \overline{X}_{2} \, \overline{X}_{3} \, \overline{X}_{4}$$
 (4-26)  

$$B (4, 2) = \overline{X}_{1} \, \overline{X}_{2} \, X_{3} \, X_{4} \, \overline{X}_{5} + \overline{X}_{1} \, X_{2} \, \overline{X}_{3} \, X_{4} \, \overline{X}_{5} + \overline{X}_{1} \, X_{2} \, X_{3} \, \overline{X}_{4} \, \overline{X}_{5}$$

$$+ X_{1} \, \overline{X}_{2} \, \overline{X}_{3} \, X_{4} \, \overline{X}_{5} + X_{1} \, \overline{X}_{2} \, X_{3} \, \overline{X}_{4} \, \overline{X}_{5} + X_{1} \, X_{2} \, \overline{X}_{3} \, \overline{X}_{4} \, \overline{X}_{5}$$

$$(4-27)$$

$$B_{U}(G) = \overline{X}_{1} \overline{X}_{2} \overline{X}_{3} + B(3, 1) + B(4, 2)$$
 (4-28)

4. 3. 3 <sub>1</sub>F<sub>1</sub>システムのB<sub>2</sub>の一般形

4.3.2節と同様にすれば,式(4-22)に対応するものとして次式が得られる.

$$R = p^{-l} \sum_{i=0}^{k-1} {l+i-1 \choose i} q^{-i}$$
 (4-29)

また、 $\overline{B}(j,r)$  は $\overline{B}(j,r)$  における $x_i$ を $\overline{x_i}$ に、 $\overline{x_i}$ を $x_i$ に変換したものとして次のように定義できる.

[定義4-5] B(i, r) は (i, j) 個の互いに素な積項の和であり、各

積項はj+1個の相異なる変数を含むものとする。その変数の内訳は、1個の $x_{j+1}$ ,  $X_j$ に含まれる相異なるx個の変数 $x_i$ の補元 $\overline{x_i}$ , および $X_j$ の残り(j-x個)の変数 $x_i$ である。

式(4-25)に対応するものとして次式が導ける.

$$B_R(F) = x_1 x_2 \cdots x_l + \sum_{i=1}^{k-1} \overline{B}(l+i-1, i)$$
 (4-30)

### 4.3.4 Heidtmann の改良 I E 法との比較

 $I_n = \{1, 2, \cdots, n\}$  において、 $I_n$ に属するi個の相異なる要素から作られる集合のクラスを $N_i$ とおく.

 $_{n}G_{k}$ システムの信頼度 R , 非信頼度 U に関して Heidtmann は次の 2 つの定理を導いている.

[定理4-3] [47] 各要素の相異なる信頼確率p (非信頼確率q) が与えられるとき、ステップi に関して次式が成立する。

$$R_{(i)} = R_{(i-1)} + (-1)^{-i-1} {\binom{i+k-2}{k-1}} \sum_{I \in N_{i+k-1}} \prod_{r \in I} P_r \quad (4-31)$$

$$U_{(i)} = U_{(i-1)} + (-1)^{-i-2k+1} {\binom{i+l-2}{l-1}} \sum_{I \in N_{i+l-1}} \prod_{r \in I} q_r$$

$$(4-32)$$

[定理4-4] [47] 各要素に均一な信頼確率p (非信頼確率q) が与えられるとき、ステップi に関して次式が成立する.

$$R_{(i)} = R_{(i-1)} + (-1)^{i-1} {\binom{i+k-2}{k-1}} {\binom{i+k-2}{k-1}} p^{i+k-1}$$
 (4-33)

$$U_{(i)} = U_{(i-1)} + (-1)^{i-2k+1} {i+l-2 \choose l-1} {i+l-1 \choose i+l-1} q^{i+l-1} \qquad (4-34)$$

 $_nG_k$ システムにおいて、改良 I E法によって得られる積項数  $S'_k$ は、式(4-31) および  $R=R_{(l)}$  より、

$$S'_{k} = \sum_{i=k}^{n} \binom{n}{i}$$
 (4-35)

と表せ、また木展開法における積項数 $S_k$ は、表4-3から次式で与えられる。

$$S_k = \sum_{l=0}^{k-1} {l+l-1 \choose l}$$
 (4-36)

式(4-36)は、

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \tag{4-37}$$

の関係を用いれば、次式のように導ける.

$$S_k = \binom{n}{k-1} = \binom{n}{k}$$
 (4-38)

一方、 $_nF_l$ システムに改良 I E法を適用するとき、その積項数  $S_l$ は式(4-32) および  $U=U_{(k)}$  から次のように示せる。

$$S'_{l} = \sum_{i=1}^{n} {n \choose i} = \sum_{i=0}^{k-1} {n \choose i}$$
 (4-39)

いま、kの値をk < n/2に限れば、次式の関係が成立することは明かである。

$$S_{k} < S_{l} < S_{k}$$
 (4-40)

式(4-40)の関係から,

- (i) Heidtmann の改良 I E 法 [47] (#1 で示す) には  $_{i}F_{i}$  システムを、
- (ii) 木展開法(#2)には $_{i}G_{k}$ システムを対象として、

n個の要素の信頼確率がすべて異なっている仮定のもとで行った実験結果を表 4-4に示す.なお,計算には九州大学大型計算機センター FACOM M-382,言語は FORTRAN を使用した.

表4-4から、木展開法は改良IE法と比較して、

- (i) 積項数が少なく、特にして指定したnに対してkがn/2に近くなる程その差が顕著となる、
- (ii) 積項1個当たりの生成が高速である, ことがいえる.

なお、k>n/2 の場合、木展開法では $_nF_l$ システムの $B_R$ (F) を算出すれば、積項数が少なくてすむことになる。

また、 $_{n}F_{l}$ システムに対してSDP法[15]の適用を試みた、その結果、例え

表4-4 実験結果

л	k	積項	数	CPU時間	(msec)
		#1	#2	#1	#2
15	4	576	455	.11	3
	6	4944	3003	67	12
20	4	1531	1140	33	7
	6	21700	15504	363	74
	8	137980	77520	1797	306
	10	431910	167960	4752	569
25	4	2626	2300	79	17
	6	68406	53130	1373	294
	8	726206	480700	11294	2180
	10	3850756	2042975	48664	7959
	12	11576916	4457400	126114	15346
30	4	4526	4060	155	36
	6	174437	142506	4123	897
	8	2804012	2035800	50172	10422
	10	22964087	14307150	334618	62808
	12	_	54627300	, ı –	211313
	14	_	119759850	_	418264

<sup>-</sup> は、演算時間の関係で実行を打ち切ったことを示す.

ば $_{15}F_{10}$ システム( $_{15}G_{6}$ システムと等価)において,3003 個の積項を得るために 49333 msec もの演算時間が費やされた.表4-4に比べて,多大な演算時間を必要とした理由は,互いに素な積項を生成するためのブール演算が 6831825 回も行われたことにある.このことは,システムの規模の増大に伴い,ブール演算の回数が指数的に増加するため, $_{n}F_{n}$ システムに $_{n}F_{n}$ システムに $_{n}F_{n}$ DP法を適用しても有意な結果が得られないことを示している.

#### 4.4 近似信頼度

近似信頼度の算出法として,次のものがある.

- (i) Heidtmann の改良 I E法[47],
- (ii) 正規近似法[1],
- (iii) Poisson 近似法[1]
- (i)は正確な信頼度計算の途中で演算を打ち切り,従来のIE法よりも信頼度のよい上限値あるいは下限値を少ない繰り返し回数(繰り返し回数の最大値n-k+1あるいはk)で求めようとするものであるが,各段階での計算量がn,kの増大とともに指数的となるため,小さなシステムにしか適用できない.(ii),(iii)は計算精度あるいは適用範囲に難点がある.

したがって、少ない手間でよい精度の近似解を求める手法の開発が必要となる。本節では、各要素の信頼確率が同一であるとき、その信頼度がO(k)で演算できることを活用したO(n)の近似アルゴリズムの構築について考察する。

### 4.4.1 木展開法による積項数と時間複雑度

以下で議論するシステムは定理 4-1 の性質を持っているので、以下では特に断らない限り。 $F_A$ システムを取り扱うものとする。

4.3節において用いられた有向木の性質について示しておこう.

有向木 I上の葉を除く任意の頂点 vにおいて、vに関する積項木 I (v) は

次の性質を持っている.

- (i) T(v) は根からvに至るパス $\mu(v)$  を含む.
- (ii) T(v) は v のすべての子を含む.
- (iii) T(v) は $\mu(v)$  上の頂点の兄をすべて含む.

以下では、vを積項木T(v)の探索頂点と呼ぶことにする。図4-2において、 $\bullet$ 印の頂点はそれぞれ探索頂点を示している。

次に、 $B_R(F)$  の積項数について示す.

 $_{n}F_{k}$ システムの信頼度表現式は式(4-16)のブール変数  $x_{i}$ を  $\overline{x}_{i}$ に、  $\overline{x}_{i}$ を  $x_{i}$ に変換すれば得られる.

$$B_{R}(F) = X_{1} X_{2} X_{3} + X_{1} X_{2} \overline{X}_{3} X_{4} + X_{1} X_{2} \overline{X}_{3} \overline{X}_{4} X_{5} + X_{1} \overline{X}_{2} X_{3} X_{4}$$

$$+ X_{1} \overline{X}_{2} X_{3} \overline{X}_{4} X_{5} + X_{1} \overline{X}_{2} \overline{X}_{3} X_{4} X_{5} + \overline{X}_{1} X_{2} X_{3} X_{4}$$

$$+ \overline{X}_{1} X_{2} X_{3} \overline{X}_{4} X_{5} + \overline{X}_{1} X_{2} \overline{X}_{3} X_{4} X_{5} + \overline{X}_{1} \overline{X}_{2} X_{3} X_{4} X_{5}$$

$$(4-48)$$

式(4-48)の $B_R(F)$  は積項の要素数に注目して次のように整理することができる。

$$B_{R}(F) = X_{1} X_{2} X_{3}$$

$$+ X_{1} X_{2} \overline{X}_{3} X_{4} + X_{1} \overline{X}_{2} X_{3} X_{4} + \overline{X}_{1} X_{2} X_{3} X_{4}$$

$$+ X_{1} X_{2} \overline{X}_{3} \overline{X}_{4} X_{5} + X_{1} \overline{X}_{2} X_{3} \overline{X}_{4} X_{5} + X_{1} \overline{X}_{2} \overline{X}_{3} X_{4} X_{5}$$

$$+ \overline{X}_{1} X_{2} X_{3} \overline{X}_{4} X_{5} + \overline{X}_{1} X_{2} \overline{X}_{3} X_{4} X_{5} + \overline{X}_{1} \overline{X}_{2} X_{3} X_{4} X_{5}$$

$$(4-49)$$

式(4-30), および各積項木T(v) はその探索頂点に1対1対応できることから,  $B_R(F)$  の積項数 $S_k$ に対して次式が成立する。なお,式(4-50)は式(4-38)と一致している。

$$S_k = \sum_{i=0}^{k-1} {i \choose i} = {i \choose k} = {i \choose k-1}$$
 (4-50)

図4-5に信頼度計算アルゴリズムを示す.このアルゴリズムは積項生成と並行して信頼度を計算するように構築したものである.

アルゴリズム UEQR では4つのスタックが使用されており、WをVの子とおくとき、

```
procedure UEQR(p_1 \sim p_n, k, R)
   begin
      l:=n-k+1
      RE(0):=1.0;
      for i:=1 until l do
          begin
             Term(i) := x_i ;
             Depth(i):=1;
             Son(i) := l-i+1;
             RE(i):=RE(i-1)*p_i
          end;
      R:=RE(i);
      while i>0 and k=1 do
          begin
             if Term(i) is not a complemented variable,
                    say x_i, and Son(i) \neq 0 then
                 begin
                    Term(i):=\overline{x}_{j};
                    RE(i):=RE(i-1)*q_i;
                    for m:=1 until Son(i) do
                       begin
                           Term(i+m):=x_{j+m};
                           RE(i+m):=RE(i+m-1)*p_{j+m};
                           Depth(i+m):=Depth(i)+1 ;
                           if Depth(i)<k-1</pre>
                              then Son(i+m):=Son(i)-m+1;
                              else Son(i+m):=0
                       end;
                    i:=i+Son(i);
                    R:=R+RE(i)
                 end;
             else i:=i-1
          end
    end UEQR;
```

図4-5 アルゴリズム UEQR

- (i) Term はT(v)から得られる積項,
- (ii) RE は積項の信頼度,
- (iii) Depth はwの深さ,
- (iv) Son はwの子の数,

を格納するようにしている.

次に, 信頼度計算の手間について考察する.

Term 上での変数の動きを T (v) の枝に置き換えて説明すると以下のようになる.

最初の for loop では T (root) を作る. このときラベルの小さいものを優先してスタックするようにしており、while loop 中の for loop でも同様である. while loop では,スタック先頭にあるラベル j の枝 (v, w) に対して,wが Tの葉でなく (Son (i)  $\neq 0$ ) かつ x ならば,x を x と置き換えた後 T (w) を作る. もしwが Tの葉ならば T (w) が存在し得ないので,T (v) 上で葉となっている頂点までバックトラックする.これは Term 上で補元でない変数を探すことに相当する.

以上の動きは Tを左から先行順探索(traverse in preorder)することと等価である。 Tの各枝に対して2つの状態 x, xが取られるから,Term 上で必要とする操作の回数は Tの枝数( $\mid T\mid$  と書く)の2倍に等しい。RE( $\cdot$ )上での演算は Term の動きに並行して行われるから,その演算回数もまた $2\mid T\mid$  に等しい。さらに,T(v)から T(w)が作られるごとに Son および Depth が v の子(w) の数だけ操作されるから,それぞれの操作回数は全体として  $\mid T\mid$  に等しい。

式(4-50)および $M=\left( \begin{smallmatrix} n \\ k \end{smallmatrix} \right)=S_{k+1}$ から,Tの枝は図4-6のように分割できる. したがって, $\mid T\mid$ は

$$|T| = S_k + S_{k+1} - 1 = S_k (n+1) / k - 1$$
 (4-51)  
となる. このことから次の定理を得る.

[定理4-5] アルゴリズム UEQR の時間複雑度は積項1 個当たり O(n/k) である.

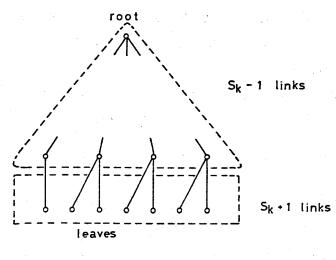


図4-6 有向木 Tの枝数

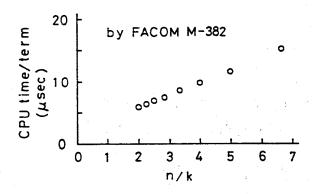


図4-7 アルゴリズム UEQR よる積項1個当たりの演算時間

積項1個当たりの演算時間を調べるため,n=20に対して行った実験結果を図4-7に示す。 $_{30}F_{12}$ システムに対してアルゴリズム UEQR を適用すれば,図4-7より積項1個当たりの演算時間が約 7  $\mu$  sec/term であり,積項数は54627300 であるので,信頼度の計算時間は約 6 分となる。なお,改良 I E 法で計算すべき積項数は上記手法の約 2 倍であり,計算時間も約 2 倍が見積もられる。したがって,いずれの算法も正確な信頼度の算出は n,k の値が小さなシステムに限られる。

また、改良 I E 法を演算の途中で打ち切って近似値を算出する算法も提案されているが[47]、 $_nG_l$ システムのステップi(ステップ総数k)における積項数は( $_{l+1-1}$ ) = ( $_{k-1}$ ) 個にもなるため、その適用範囲は $_n$ , kによって制限を受ける。

近似計算法は適用範囲( $\bar{p} \le 0.5$  でかっ n  $\bar{p} \ge 5.0$ ,あるいは  $\bar{p} > 0.5$  でかっ n  $\bar{q} > 5.0$ )が広いものの,表4-5のように精度に関して難点がある.また, Poisson 近似法は適用範囲が狭い( $\bar{p} \le 0.05$  でかっ  $n \ge 2$  0 あるいは  $\bar{q} \le 0.05$  でかっ  $n \ge 2$  0 ).ここで,  $\bar{p}$  (=1.0  $-\bar{q}$ ) は与えられた信頼確率  $p_1 \sim p_n$  の平均値を示す.

なお,表4-5および以下の実験において,与えられたデータの代表的表現として要素の信頼確率の最小値  $p_{\min}$ ,最大値  $p_{\max}$ ,平均値  $\bar{p}$ ,および標準偏差  $\sigma$  を用いることにする.

## 4.4.2 近似計算法

 $_nF_k$ システムにおいて、すべての要素が同一の信頼確率  $_P$  (= 1.0-  $_q$ ) を持つとき、式(4-49)から信頼度  $_R$ は次式で与えられる.

$$R(n, p, k) = p \sum_{i=0}^{l} {i \choose i} q^{i}$$
 (4-52)

式(4-52)を用いた信頼度計算アルゴリズムを図4-8に示す。p lは EXP ( l\* LOG (p) ) と展開して計算され,関数 EXP および LOG の演算時間が計算機の

表4-5  $_{20}F_{A}$ システムに対する正規近似法の適用例  $(P_{min}=0.4, P_{max}=0.6, \bar{p}=0.49518, \sigma=0.06085)$ 

	正確な信頼度	正規近似法に よる信頼度	誤差 R – R
k	R	R.	$\frac{R-R}{R} \times 100.0$
2	1.49×10 <sup>-6</sup>	2.88×10 <sup>-7</sup>	5055.6
4	1.04×10 <sup>-8</sup>	9. 43×10 <sup>-5</sup>	1003. 1
6	1.79×10 <sup>-2</sup>	6.78×10 <sup>-8</sup>	164. 6
8	1. 21×10 <sup>-1</sup>	1. 14×10 <sup>-1</sup>	5. 57
10	3.94×10 <sup>-1</sup>	5. 24×10 <sup>-1</sup>	-24.8
12	7.36×10 <sup>-1</sup>	9. 08×10 <sup>-1</sup>	-18.9
14	9. 39×10 <sup>-1</sup>	9. 95×10 <sup>-1</sup>	-5. 68
16	9.94×10 <sup>-1</sup>	9.99×10 <sup>-1</sup>	-0.626
18	9. 99×10 <sup>-1</sup>	9. 99×10 <sup>-1</sup>	-0.021

図4-8 アルゴリズム EQR

表4-6 R(n,0.9, k)の演算時間

(-)	l	10
(a)	и —	10

П	演算時間(μ sec)
20	15
40	16
60	16
80	16
100	16

(b) n = 100

k	演算時間(μ sec)
10	16
20	27
30	39
40	50
50	62

語長に依存することに注意すれば[53], アルゴリズム EQR の時間複雑度が O(k) であることは明らかであり, 表4-6の R(n, 0.9, k) の実験結果に もそのことが示されている.

以下では、アルゴリズム EQR を用いた近似アルゴリズムについて議論する。 [定義4-6] 相異なる信頼確率 $P_1 \sim P_n$ を持つ。 $P_k$ システムが与えられたとき、

 $R(p_1 \sim p_n; k) = R(n, p_{sk}, k)$  (4-53) を満たす $p_{sk}$ および信頼確率 $p_{sk}$ を持つシステムを、それぞれ与えられたシステムの擬似信頼確率および擬似システムと名づける。

定義4-6における $p_{ak}$ とkの関係を調べるため,R ( $p_1 \sim p_{20}$ ; k),k = 1,2,…,20をそれぞれ計算し,擬似信頼確率 $p_{ak}$ を求める実験を行った.実験結果のうち,2種類のデータ(データ1: $p_{min}=0.2$ , $p_{max}=0.8$ , $\bar{p}=0.48553$ , $\sigma=0.18256$ , データ2: $p_{min}=0.4$ , $p_{max}=0.6$ , $\bar{p}=0.49518$ , $\sigma=0.06085$ )に対して得られた $p_{ak}$  (k=1, 2, …, 20)をプロットしたのが図4-9である.〇印はデータ1,×印はデータ2に対するもので,いずれも直線近似が可能であることが分かる.なお,データ2は表4-5で用いたデータと同一である.

本節で提案する近似信頼度計算法は,擬似信頼確率 $P_{ak}$ の直線性を利用するものであり,k=1 およびk=n における信頼度から $P_{a1}$ ,  $P_{an}$ をそれぞれ計算し, $P_{a1}$ と $P_{an}$ の値から与えられたk (k=2, 3, …, n-1) に関する近似信頼確率 $P_{a}$ を推定し,近似信頼度R (n,  $P_{a}$ , k) を計算するものである。図4-10に示す近似アルゴリズムの時間複雑度は, $P_{a1}$ および $P_{an}$ の算出に支配される\*.

近似アルゴリズム APR に関する実験によれば、演算時間はkに依存せずO(n)である(図4-11参照).

<sup>\*</sup>  $P_{s_1}$ は EXP ( LOG ( $P_1*P_2* \cdots *P_n$ )  $/ \pi$ ) と展開して計算が行われ,その演算時間は $\pi$ 個の $P_i$ の積に依存する。また, $P_{s_n}$ の演算時間についても同様である。

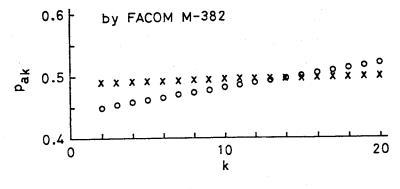


図4-9 Pak- k特性

```
\begin{array}{c} \text{procedure APR}(p_1{\sim}p_n,k,R_a) \\ \text{begin} \\ p_{a1} \colon = (p_1{*}p_2{*}\cdots{*}p_n)^{1/n} \; ; \\ p_{am} \colon = 1.0{-}(q_1{*}q_2{*}\cdots{*}q_n)^{1/n} \; ; \\ p_a \colon = p_{a1}{+}(p_{an}{-}p_{a1}){*}(k{-}1)/(n{-}1) \; ; \\ \text{EQR}(n,p_a,k,R_a) \\ \text{end APR} \; ; \end{array}
```

図4-10 アルゴリズム APR

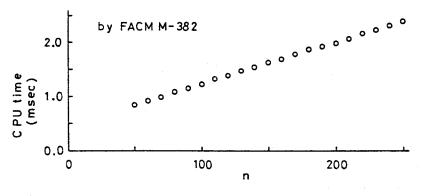


図4-11 アルゴリズム APR の演算時間

なお、計算機の記憶部の語長は有限であるため、信頼確率の与え方によっては $P_{s1} = \begin{pmatrix} P_1 P_2 & \cdots & P_n \end{pmatrix}^{1/n}$ あるいは $P_{sn} = 1.0 - \begin{pmatrix} q_1 q_2 & \cdots & q_n \end{pmatrix}^{1/n}$ の演算が不可能となる場合がある。

計算機実験には、九州大学大型計算機センター FACOM M-382 (プログラム用言語として FORTRAN を用いた)を使用し、また実数計算には8バイト変数(仮数部57ビット、指数部7ビット、仮数部にはサインビットを1ビット含む)を用いた。

#### 4.4.3 誤差の評価

ここでは、図4-10のアルゴリズムによって算出される近似信頼度に含まれる 誤差について考察する。

まず,擬似信頼確率P。kと近似信頼確率P。の偏差を $\triangle P$ と仮定すれば,式 (4-52) は次のように書ける.

$$R$$
 ( $n$ ,  $p_{ak}$ ,  $k$ )

$$= (p_{s} + \triangle p) {l \sum_{i=0}^{k-1} {l \choose i}} (q_{s} - \triangle p)^{i}$$
 (4-54)

いま,  $l \triangle P \ll P$   $_a$ と仮定すれば,式(4-54)は次のように展開できる.なお,表示を簡単にするため,正確な信頼度 R (n, P  $_a$ , k) および近似信頼度 R (n, P  $_a$ , k) をそれぞれ R, R  $_a$ で表すものとする.

$$R = (p_s + l \triangle p_{p_s}^{l-1})$$

• 
$$\sum_{i=0}^{k-1} {i \choose i} (q_a^{i} - i \triangle p q_a^{i-1})$$

$$-p_{s}! \triangle p_{i=0}^{k-1} (i^{l+i-1}) i q_{s}^{i-1}$$

$$= R_{s} + l \triangle p R_{s} / p_{s} - p_{s}^{l} \triangle p \sum_{i=1}^{k-1} {i \choose i} i q_{s}^{i-1}$$
 (4-55)

式(4-55)から誤差の評価式として次の関係を得る.

$$\frac{R-R_{a}}{R_{a}} = \frac{l \triangle P}{P_{a}} - \frac{P_{a}^{l} \triangle P}{R_{a}}$$

$$\cdot \sum_{i=1}^{k-1} {\binom{l+i-1}{i}} i q_{a}^{i-1} \qquad (4-56)$$

式(4-56)に基づく誤差評価アルゴリズムを図4-12に示す.

なお、kにおける偏差を $\Delta P_k$ とおくとき、 $P_k$ とおくとき、 $P_k$ とかった。 $P_k$ とかった。 $P_k$ は変化するが、これまでの実験結果によれば、 $P_k$ の最大値が  $P_k$ 0 を越えることはなく、また  $P_k$ 0 の最大値が  $P_k$ 1 の値が  $P_k$ 2 付近にあるとき  $P_k$ 1 は最大値を持つことが多かった。したがって、 $P_k$ 2 において $P_k$ 2 において  $P_k$ 3 におけば、誤差の推定においてその上限値を示すことができよう。 $P_k$ 4 ( $P_k$ 4 におけば、誤差の推定においてその上限値を示すことができよう。 $P_k$ 5 ( $P_k$ 6 における。 $P_k$ 7 を満たす  $P_k$ 8 の算出において、 $P_k$ 9 の算出が不可能となることがある。したがって、誤差評価においては  $P_k$ 9 の算出が不可能となることがある。したがって、誤差評価においては  $P_k$ 9 の算出が不可能となることがある。したがって、誤差評価においては  $P_k$ 9 の算出が不可能となることがある。したがって、誤差評価においては  $P_k$ 9 の算出が不可能となることがある。したがって、誤差

## 4.4.4 近似アルゴリズムの適用範囲

誤差評価アルゴリズム %ERROR に対して行った実験結果を表4-7, 4-8に示す。 ここで,Rは正確な信頼度,R。は近似信頼度,E2, E4はその ERROR の真値 (=(R-R))/R0× 100.0),% Eはアルゴリズム %ERROR による誤差の 推定値(式(4-56)の百分率)をそれぞれ表している。なお,表4-8はP04-44特性を示すために用いた 2種類のデータ(図4-9)に関する実験結果である。

また、同一システムに対して、k=n/2とおいて行った誤差の推定実験では、%Eが 2.0 %を越えることはなかった。

次に、標準偏差が大きいデータ(n=20,  $p_{\min}=0.0$ ,  $p_{\max}=1.0$ ,  $\bar{p}=0.47588$ ,  $\sigma=0.30427$ ) の実験結果を表4-9に示す。 $\sigma$ が大きいとき,  $p_{\alpha 1}$ と $p_{\alpha n}$ を結ぶ直線に対して, $p_{\alpha k}$ - k特性が凸型あるいは凹型の曲線にな

```
procedure %ERROR
    begin
         UEQR(p_1 \sim p_n, 2, R);
         find p_{\alpha 2} such that R(p_1 \sim p_n; 2) = R(n, p_{\alpha 2}, 2);
         APR(p_1 \sim p_n, 2, R);
         \Delta p := p_{a2} - p_a; l := n-k+1;
          if k>2 then
              begin
                   p_{\alpha} \colon = p_{\alpha 1} + (p_{\alpha n} - p_{\alpha 1}) * (k-1)/(n-1) ;
                   \Delta p := \alpha * \Delta p
              end;
          q_a := 1.0 - p_a;
          S:=\Delta S:=l;
          for i:=2 until k-1 do
               begin
                   \Delta S:=\Delta S*q_{a}*(l+i-1)/(i-1);
                   S:=S+AS
               end;
          EQR(n,p_{\alpha},k,R_{\alpha})
          E:=\Delta p*(l/p_{\alpha}-S*p^{l}/R_{\alpha})*100.0
      end %ERROR;
             図4-12 アルゴリズム %ERROR
```

表4-7 "F2システムの近似信頼度と誤差

(a)  $p_{min} = 0.8$ ,  $p_{max} = 0.9$ 

Л	<del>,</del>	σ	Ra	E 2	% E
50 100 150 200	0.8550 0.8481 0.8525 0.8461	0. 02707 0. 02965 0. 02914 0. 02836	$3.70 \times 10^{-3}$ $1.25 \times 10^{-6}$ $9.98 \times 10^{-10}$ $1.02 \times 10^{-13}$	0. 0142 0. 3775 0. 3836 0. 3533	0.0142 0.3768 0.3829

-は,k=2 における擬似信頼確率が 算出できなかったことを示す.

(b)  $P_{min} = 0.9$ ,  $P_{max} = 0.99$ 

л	Īp,	σ	R.	E 2	% E
50 100 150 200 250	0. 9495 0. 9433 0. 9473 0. 9415 0. 9466	0.02436 0.02669 0.02623 0.02552 0.02622	2. 73×10 <sup>-1</sup> 1. 98×10 <sup>-2</sup> 2. 62×10 <sup>-3</sup> 7. 25×10 <sup>-5</sup> 1. 53×10 <sup>-5</sup>	-0. 0109 0. 5984 0. 6496 0. 5767 0. 6695	-0.0109 0.5967 0.6475 0.5751 0.6673

表4-8 20 F kシステムの近似信頼度と誤差

(a)  $p_{\text{min}} = 0.2$ ,  $p_{\text{max}} = 0.8$ ,  $\bar{p} = 0.48553$ ,  $\sigma = 0.18256$ 

k	R	R .	$E_{k}$	% E
2	3.3645×10 <sup>-6</sup>	3. 4261×10 <sup>-6</sup>	-1.7978	-1.8132
4	3.8557×10 <sup>-4</sup>	4.0024×10 <sup>-4</sup>	-3.6654	-7. 3854
6	9.8721×10 <sup>-3</sup>	1.0265×10 <sup>-2</sup>	-3.8296	-6.0257
8	8.9066×10 <sup>-2</sup>	9. 1797×10 <sup>-2</sup>	-2.9747	-4. 9832
10	3. 5128×10 <sup>-1</sup>	3. 5753×10 <sup>-1</sup>	-1.7456	-4. 2201
12	7. 1937×10 <sup>-1</sup>	7. 2443×10 <sup>-1</sup>	-0.6981	-3.6249
14	9. 4047×10 <sup>-1</sup>	9.4189×10 <sup>-1</sup>	-0.1507	-2.9977
16	9.9496×10 <sup>-1</sup>	9.9508×10 <sup>-1</sup>	-0.0120	-2. 2007
18	9.9987×10 <sup>-1</sup>	9.9988×10 <sup>-1</sup>	-0.0002	-1. 3130

(b)  $p_{\text{min}} = 0.4$ ,  $p_{\text{max}} = 0.6$ ,  $\bar{p} = 0.49518$ ,  $\sigma = 0.06085$ 

k	R	R.	$E_{k}$	% E
2	1. 4866×10 <sup>-5</sup>	1. 4873×10 <sup>-5</sup>	-0.0481	-0.0481
4	1.0399×10 <sup>-3</sup>	1.0409×10 <sup>-3</sup>	-0.0989	-0.1958
6	1.7927×10 <sup>-2</sup>	1.7946×10 <sup>-2</sup>	-0.1047	-0.1637
8	1. 2068×10 <sup>-1</sup>	1. 2078×10 <sup>-1</sup>	-0.0832	-0.1383
10	3. 9417×10 <sup>-1</sup>	3. 9437×10 <sup>-1</sup>	-0.0507	-0.1197
12	7. 3581×10 <sup>-1</sup>	7.3597×10 <sup>-1</sup>	-0.0217	-0.1044
14	9.3865×10 <sup>-1</sup>	9. 3870×10 <sup>-1</sup>	-0.0053	-0.0872
16	9. 9368×10 <sup>-1</sup>	9.9369×10 <sup>-1</sup>	-0.0005	-0.0649
18	9.9979×10 <sup>-1</sup>	9.9979×10 <sup>-1</sup>	-0.00001	-0.0393

表4-9 標準偏差が大きい20 F x システムの近似信頼度と誤差

 $(P_{\min} = 0.0, P_{\max} = 1.0, \bar{p} = 047588, \sigma = 0.30427)$ 

k	R	$R_{a}$	E ,	% E
2	4. 49×10 <sup>-8</sup>	2.96×10 <sup>-8</sup>	51. 3	41. 9
4	2.81×10 <sup>-5</sup>	2. 25×10 <sup>-4</sup>	25.1	162.3
6	2. 25×10 <sup>-3</sup>	2. 26×10 <sup>-3</sup>	-0.43	124.5
8	4. 49×10 <sup>-2</sup>	5.00×10 <sup>-2</sup>	-10.2	95. 9
10	2.87×10 <sup>-1</sup>	3. 18×10 <sup>-1</sup>	-9.97	76.1
12	7. 17×10 <sup>-1</sup>	7. 57×10 <sup>-1</sup>	-5. 18	62.7
14	9. 59×10 <sup>-1</sup>	9. 70×10 <sup>-1</sup>	-1.12	50.2
16	9. 98×10 <sup>-1</sup>	9.99×10 <sup>-1</sup>	-0.06	35.0
18	9. 99×10 <sup>-1</sup>	9.99×10 <sup>-1</sup>	-0.0004	20.0

ったり、また k=n/2 付近において変曲点を持つ曲線となることがあるため、ERROR の変動幅が大きくなることがある。さらに、このデータでは $\Delta p$  が大きいため、式(4-56)の成立条件である  $l \Delta p \ll p$  が満たされず、誤差推定そのものの信頼性が失われている。

以上から、次のことが言える.

- (i) 提案した近似アルゴリズムの時間複雑度はO(n)であり、要素数 250 のシステムでも約 2.0 msec で信頼度を計算できる.
- (ii) 標準偏差が 0.1 以下である場合,近似アルゴリズムは実用性のある 近似信頼度を算出することができる.
- (iii) 近似アルゴリズムの適用範囲は、PalおよびPanが計算可能かどうかに依存している。なお、実数計算に8バイト変数を用いる場合、要素数 250 程度までのシステムに適用可能である。
- (iv)  $R(p_1 \sim p_n; 2) = R(n, p_{s2}, 2)$  を満たす $p_{s2}$ が計算可能であれば、誤差の推定を行うことができる.

#### 4.5 結言

4. 3 節においては,k-o u t -o f -n システムの正確な信頼度(非信頼度)表現式を導く木展開法の原理およびアルゴリズムについて述べ,さらに Heidtmann の改良 I E 法との理論的および実験的比較を行った.

得られた結果は以下のとおりである.

- (i) 木展開法は、表現式が互いに素な積項の和形式で得られ、かつ積項 数が少ない。
- (ii) 積項の生成に関して、Heidtmann の改良 I E 法は要素の可能な組み合せを求める必要があるが、木展開法はスタック上の操作だけでよい。さらに、4.4節においては、与えられた。 $F_k$ システムに対して、 $R(p_1 \sim p_n; k) = R(n, p_{ak}, k)$  を満たす擬似信頼確率 $p_{ak}$ を求めるとき、 $p_{ak} k$ 特性が直線近似で表現できるという性質を利用したO(n) の近似計算アルゴリズムを提案した。さらに、近似アルゴリズムの誤差の推定法を述べ、

種々の実験から提案したアルゴリズムの有効性を示した.

なお、このアルゴリズムは与えられたデータの標準偏差 $\sigma$ が 0.1 以下の場合に対してのみ有効であり、 $\sigma > 0.1$  の場合における近似アルゴリズムの開発が今後に残された課題である。

# 第5章 結論

本研究で得られた成果と残された問題点を総括して述べる。

第2章では、ネットワークのs-t信頼度(sからtに至る正常な枝だけから構成されるs-tパスが少なくとも1つ存在する確率)を求める3つのグラフ理論的方法、

- (i) s-t アサイクリックグラフを生成するパス付加法、
- (ii) 木展開法.
- (iii) パスの処理順序と積項数の関係(改良木展開法), について述べた。

理論的検討および実験結果より、枝数の少ないパス (短いパス) から優先的に処理する改良木展開法は、得られる表現式の積項数が最も少なく、かつその算出時間も少ない、という結論が得られた.しかしながら、得られる表現式が因数分解された形式でないため、表現式内には同一のリテラルが数多く含まれ、また積項数も多い.したがって、信頼度表現式を導くアルゴリズムの取り扱えるネットワークのサイズはまだ小さい.因数分解された表現式を直接得るグラフ理論的方法、および精度のよい近似信頼度を得る算法、に対する検討が今後の課題である.

第3章では、ネットワークのSAT信頼度(ソースからソース以外のすべての頂点に至る正常な枝による経路が少なくとも1つ存在する確率)を求めるグラフ理論的方法について述べた、これに関して、

- (i) 木生成法,
- (ii) アサイクリックサブグラフ生成法,
- (iii) 最小の積項数を求める算法(セクショングラフ法), を提案した.

この三つの算法のうちでセクショングラフ法が最も高速であり、かつ得られる信頼度表現式に含まれるリテラル数も最小となるが、表現式内には同じリテラルを含むことがある(例えば、式(3-38)の第1項にはx<sub>2</sub>、第2項にはx<sub>3</sub>が

それぞれ2個含まれ、これらは因数分解可能である). これを解消して、更に 簡潔な表現式を得ることが必要である.

第4章では、k-out-of-nシステムの信頼度算出法について述べた。 このシステムの信頼度算出に関して、

- (i) 正確な信頼度を求める木展開法,
- (ii) 近似信頼度の計算法,

#### を提案した.

木展開法は、スタック上の操作だけで正確な信頼度表現式を生成できるため、既存の算法よりも高速である。また、擬似信頼確率の直線性を利用した近似計算アルゴリズムは、標準偏差が 0.1 以下である場合、要素数 250 までのシステムの信頼度が計算でき、また得られる信頼度の実用性にも問題ないことを示した。しかしながら、近似アルゴリズムは与えられたデータの標準偏差が 0.1 以下の場合に対してのみ有効であり、標準偏差が 0.1 以上の場合に適用可能な近似アルゴリズムの開発が残された課題である。

最も一般的なネットワーク信頼度問題は、ネットワーク上の指定した頂点sからそれ以外の複数個の頂点の集合Kへの到達可能性を問題とするSMT信頼度(source-to-multi-terminal reliability)問題であり、また最も取扱いの困難な問題でもある。s-t信頼度,SAT信頼度を求める問題は、SMT信頼度問題の特別なクラスの一つである。本論文で提案した各種算法をSMT信頼度問題に適用することは難しく、新たな算法の開発が必要である。また、ネットワーク信頼度問題はいずれもNP困難な問題のクラスに属しているため、実用性のある信頼度を効率よく算出できる近似アルゴリズムの重要性が、今後ますます増大していくと考えられる。

## 謝辞

本研究の全過程を通じて御指導,御配慮をいただきました愛媛大学工学部電 気電子工学科有吉弘教授,直接理解ある御指導,御鞭撻をいただきました大阪 大学工学部情報システム工学科白川功教授に心から感謝する.

本研究に関し御討論,御審査をいただきました大阪大学工学部情報システム 工学科寺田浩詔教授,電子工学科児玉慎三教授,大型計算機センター熊谷貞俊 教授に厚く御礼申し上げる.

愛媛大学工学部学生,大学院生の期間を通じて御指導,御教示賜った愛媛大学工学部相原恒博教授,故奥田一郎教授に深謝する.

筆者の所属する愛媛大学工学部電気電子工学科において、御助言、御協力いただいた渡辺健二講師、山本武彦技官に御礼申し上げる.

筆者が内地研究員として白川研究室に出向以来種々の面で御援助、御協力いただいた大阪大学基礎工学部荒木俊郎助教授、工学部通信工学科馬場口登講師、工学部情報システム工学科石浦菜岐佐講師、中道和則助手(現在、大阪電気通信大学大学院学生)、猿山幸香嬢、古野電気(株)システム機器事業部近藤仁志氏、奈良工業高等専門学校情報工学科世古忠助教授、大学院学生の康敏燮氏(現在、韓国電子通信研究所半導体設計部)、重弘裕二(現在、情報システム工学科助手)、雀淏鎔氏、孫曉軍氏(現在、ダイキン工業(株))、撫原恒平氏(現在、日本電気(株))、尾上孝雄氏(現在、情報システム工学科助手)、佐藤和良氏(現在、ケーシーエス(株))、研究生の揚世宗氏(現在、大学院学生)に感謝する。特に、留学生諸氏には研究のみならず文化、生活など種々の面において興味ある多くの知見を得た。ここに記して感謝する次第である。

最後に,本研究の計算機実験に御協力いただいた愛媛大学工学部電気工学科 卒業生諸氏に感謝の意を表する.

# 参考文献

- [1] M.L. Shooman: Probabilistic Reliability: An Engineering Approach, McGraw-Hill, New York(1968).
- [2] A. Satyanarayan and A. Prabhakar: "New topological formula and rapid algorothm for reliability analysis of complex networks", IEEE Trans. Reliability, Vol. R-27, No. 2, pp. 82-100(1978).
- [3] R. R. Willie: "A theorem concerning cyclic directed graphs with applications to network reliability", Networks, No. 10, pp. 71-78(1980).
- [4] 東山,有吉:"s-tアサイクリック部分グラフの算出アルゴリズム", 電子通信学会技術研究報告,CAS82-77,pp.53-60(1982).
- [5] 有吉,東山: "有向スターポリゴンの最小帰還枝集合について",電子 通信学会論文誌(A), Vol. J64-A, No. 8, pp. 637-644 (1981).
- [6] 東山,賀上,有吉: "最小FAS決定可能なDSPのクラスについて", 電子通信学会技術研究報告,CAS81-48,pp.7-14 (1981).
- [7] 永井, 有吉: "グラフィックデータの生成法", 昭和50年電気関係学会 四国支部連合大会, pp. 21-22 (1975).
- [8] 東山,有吉: \*\* 木展開法による新しいネットワーク信頼度計算法\*\*,電子通信学会技術研究報告,CAS83-113,pp.23-28(1983).
- [9] M.O.Locks: Recursive disjoint products inclusion-exclusion, and min-cut approximations, IEEE Trans. Reliability, Vol. R-29, No. 5, pp. 368-371(1980).
- [10] M.O.Locks: Recursive disjoint products: A review of three algorithms, IEEE Trans. Reliability, Vol. R-31, No. 1, pp. 33-35(1982).
- [11] 東山,有吉: "s-tアサイクリックグラフの算出アルゴリズム",電 子通信学会論文誌(A), Vol. J67-A, No. 4, pp. 367-374(1984).

- [12] P. M. Lin, B. J. Leon and T. C. Huang: "A new algorithm for symbolic system reliability analysis", IEEE Trans. Reliability, Vol. R-25, No. 1, pp. 2-15(1976).
- [13] L. Fratta and U. G. Montanari: "A boolean algebra method for computing the terminal reliability in a communication network", IEEE Trans. Circuit Theory, Vol. CT-20, No. 3, pp. 203-211(1973).
- [14] K. K. Aggarwal, K. B. Misra and J. S. Gupta: A fast algorithm for reliability evaluation, IEEE Trans. Reliability, Vol. R-23, No. 1, pp83-85(1975).
- [15] J.A. Abraham: An improved algorithm for network reliability, IEEE Trans. Reliability, Vol. R-28, No. 1, pp. 58-61(1979).
- [16] E. Hansler, G. K. McAuliffe and R. S. Wilkow: Exact calculation of computer network reliability, Networks, No. 4, pp. 95-112(1974).
- [17] 有吉,東山: "ネットワーク信頼度評価のグラフ論的手法",電子通信 学会技術研究報告,IT83-31, pp.7-12(1983).
- [18] 東山,有吉: "ネットワーク信頼度評価のグラフ論的手法",愛媛大学 工学部紀要, Vol. 11, No. 1, pp. 43-53(1986).
- [19] 東山,向井,渡,有吉:"ネットワーク信頼度計算法の効率評価について",昭和59年電子通信学会総合全国大会,No.48(1984).
- [20] H. Ariyoshi and Y. Higashiyama: "New method for computing exact terminal reliability using a tree expansion technique", Proc. of 1984 International Symposium on Circuits and Systems, pp. 1135-1138 (May 1984).
- [21] 東山,有吉: \*\* 木展開法による新しいネットワーク信頼度計算法\*\*,電子通信学会論文誌(A), Vol. J68-A, No. 2, pp. 122-129(1985).
- [22] 東山,向井,有吉: "信頼度表現式の積項数削減の一手法",電子通信 学会技術研究報告,CAS85-19, pp. 53-59(1985).
- [23] 東山, 有吉: "信頼度表現式の積項数削減の一手法", 電子通信学会論 文誌(A), Vol. J69-A, No. 5, pp. 560-566(1986).

- [24] A. Satyanarayana and J. N. Hagstrom: A new algorithm for the reliability analysis of multi-terminal networks, IEEE Trans. Reliability, Vol. R-30, No. 4, pp. 325-334(1981).
- [25] M.O.Locks: Recent developments in computing of system-reliability, IEEE Trans. Reliability, Vol. R-34, No. 5, pp. 425-436(1985).
- [26] 野田,東山,有吉: \*\* 木によるMulti-terminalシステムの信頼度表現式に関する一考察\*、昭和62年電気関係学会四国支部連合大会,1-4(1987).
- [27] R. Johnson: "Network reliability and acyclic orientations", Networks, No. 14, pp. 489-505(1984).
- [28] R.K.Wood: "Factoring algorithms for computing K-terminal network reliability", IEEE Trans. Reliability, Vol. R-35, No. 3, pp. 269-278(1986).
- [29] M.O. Ball and G. Nemhauser: Matroids and a reliability analysis problem. Mathematics of Operations Research, No. 4, pp. 132-143 (1979).
- [30] 伊理,白川,梶谷,篠田 他:"演習グラフ理論",コロナ社(1983).
- [31] 石岡,湊,東山,有吉:"DP法の高速化について",昭和62年電気関係学会四国支部連合大会,1-1(1987).
- [32] 野田,東山,有吉,白川: Multi-terminalシステムのカットセット列挙アルゴリズム",昭和61年電気関係学会四国支部連合大会,1-7,(1986).
- [33] 東山,野田,有吉,白川:"積項生成におけるカットセットの処理順序 ",電子情報通信学会技術研究報告,CAS86-179,pp39-45(1987).
- [34] 東山,有吉,白川: "積項生成におけるカットセットの処理順序と演算 量の軽減効果",愛媛大学工学部紀要,Vol.11,No.4,pp95-103(1989).
- [35] R. C. Read and R. E. Tarjan: Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, Networks, No. 5, pp. 237-252(1975).

- [36] E. M. Reingold, J. Nievergelt and N. Deo: Combinatorial Algorithms: Theory and Practice, Prentice-Hall (1977).
- [37] G. J. Minty: "A simple algorithm for listing all the trees of a graph", IEEE Trans. Circuit Theory, Vol. CT-12, No. 1, p. 120(1965).
- [38] 野田,東山,有吉,白川: \*\*\* 木を用いたソース・全端子信頼度表現式の 最適生成法\*\*,電子情報通信学会技術研究報告,CAS87-205,pp.55-60 (1987).
- [39] Y. Higashiyama, T. Noda, H. Ariyoshi and I. Shirakawa: "A new topological method for symbolic reliability analysis of source-to-all-terminal networks", Proc. of the 27th SICE Annual Conference, pp. 911-914(1988-8).
- [40] Y. Higashiyama, T. Noda, H. Ariyoshi and I. Shirakawa: "minimizing symbolic reliability formula of source-to-all-terminal networks", Proc. of 1988 Joint Technical Conference on Circiuts/Systems. pp. 68-72(1988-11).
- [41] 野田,東山,有吉,白川: \*\*\* 木を用いたソース・全端子信頼度表現式の 最適生成法\*\*,電子情報通信学会論文誌(A), Vol. J72-A, No. 2, pp. 381-389(1989).
- [42] Y. Higashiyama, T. Noda, H. Ariyoshi and I. Shirakawa: "An optimal symbolic reliability formula using spanning trees for source-to-all-terminal systems", Memoirs of the Faculty of Engineering, Ehime University, Vol. 12, No. 1, pp. 117-132(1990).
- [43] M.O.Ball: "Computational complexity of network reliability analysis: An overview", IEEE Trans. Reliability, Vol. R-35, No. 4, pp. 230-239(1986).
- [44] Y. Higashiyama, H. Ariyoshi and I. Shirakawa: "An algorithm for genearting a factored symbolic reliability formula of directed source-to-all-terminal network", Proc. of 1989 International Conference on Circuits and Systems, pp. 774-777 (1989).

- [45] Y. Higashiyama, H. Ariyoshi, I. Shirakawa, and S. Ohba: "A method for generating a factored SAT symbolic reliability formula".

  Proc. of 1990 Joint Technical Conference on Circuits/ Systems, pp. 333-337(1990).
- [46] W.S. Meisel: Reliability in digital systems with asymmetrical failure modes, IEEEE Trans. Reliability, Vol. R-18, NO. 2, pp. 74-75(1969).
- [47] K.D. Heidtmann: "Improved method of inclusion-exclusion applied to k-out-of-n systems", IEEE Trans. Reliability, Vol. R-31, No. 1, pp. 36-40(1982).
- [48] 東山,有吉: "k-out-of-n システムへの木展開法の応用",昭和58年電 気関係学会四国支部連合大会,1-3(1983).
- [49] 東山,渡,有吉: "k-out-of-n システムへの木展開法の応用",電子通信学会技術研究報告,CAS84-72,pp.79-85(1984).
- [50] 野崎,野下:"アルゴリズムの設計と解析I",サイエンス社(1978).
- [51] 東山,有吉: "k-out-of-n システムへの木展開法の応用",電子通信学会論文誌(A), Vol. J68-A, No. 6, pp. 594-601(1985).
- [52] Y. Higashiyama and H. Ariyoshi: Tree-expansion technique applied to k-out-of-n systems. Proc. of 1985 International Symposium on Circuits and Systems, pp. 1651-1654(1985-6).
- [53] 森,名取,鳥居:"数値計算",岩波講座,情報科学-18,岩波書店 (1982).
- [54] 東山,渡,有吉: "k-out-of-n システムの近似信頼度の一計算法",電 子通信学会技術研究報告, CAS85-136, pp. 33-40(1986).
- [55] 東山,渡,有吉: "k-out-of-n システムの近似信頼度算出アルゴリズム",電子情報通信学会論文誌(A), Vol. J70-A, No. 2, pp. 212-219(1987).
- [56] Y. Higashiyama, T. Watari and H. Ariyoshi: "An algorithm for computing approximate reliability of k-out-of-n systems", Proc. of the 25th SICE Annual Conference, pp. 1059-1062(1986-7).