

自己診断システムにおけるネットワーク構造 と計算複雑さの関係

広島大 工 阿江 忠
大崎 重義

1. まえがき

分散形計算機システムの各ユニットを節点に，互いのユニット間の検査を有向枝に対応させると有向グラフ（ネットワークと呼ぶ）が得られる。各ユニット u_i は， u_j へ向かう枝の重みとして与える。

- ① ユニット u_j を検査可能である and/or
② “ u_j によって検査されることが可能
である

とある。 u_i が u_j を「故障」と判定すれば 1，「正常」と判定すれば 0 を u_i から u_j へ向う枝の重みとして与える。すべての枝の重みを要素とするベクトルを シンδροームという。シンδροームから真に故障の疑いの強いユニット群を指摘するのが，自己診断システムにおける主たる目的である。ここでネットワークと称している自己診断のためのモデルは Preparata et al.⁽¹⁾ 以来，いろいろ議論されて

されており、実用のための変形モデルも数多いし、診断の仕方
もいろいろ提案されてきている⁽²⁾。本稿では、もっとも基
本的な診断である“1回の検査から得られるシンドロームよ
りあべての故障ユニットを識別する同時故障診断⁽¹⁾”に必要
な計算の手数を議論する。

この問題が、一般には、NP-complete であることがあ
るに指摘されているので⁽³⁾⁽⁴⁾⁽⁵⁾、我々は

- ① NP-complete であるより小さいクラス、と
- ② 多項式時間で解けるでさえだけ大きいクラス
をネットワークの構造で規定する試みを示す。

2. 定義

ネットワーク N を有向グラフ $G=(V, E)$, ここに V はユ
ニットの集合 $\{u_1, \dots, u_n\}$ に対応する, とあらわす。

$(u_i, u_j) \in E$ の重みを c_{ij} とあるとき

$$c_{ij} = \begin{cases} 0 & \text{if } u_i \text{ が } u_j \text{ を "正常" と判定} \\ 1 & \text{if } u_i \text{ が } u_j \text{ を "故障" と判定} \\ \text{undefined}^+ & \text{otherwise.} \end{cases}$$

である。

故障ユニットの集合 F は次の2つの条件を満たすものとする。

+) この場合 枝 (u_i, u_j) そのものが定義されない。

条件 1 $u_i \in \bar{F} = V - F$, $u_j \in F$ なるあべでの
 $(u_i, u_j) \in E$ に対して $c_{ij} = 1$

条件 2 $u_i, u_j \in \bar{F}$ なるあべでの $(u_i, u_j) \in E$
 に対して $c_{ij} = 0$

このような $F \subseteq V$ を 無矛盾故障集合 という。

計算複雑さについてはチューリング機械のそれを用いる。
 (6) の定義にしたがって用いる。

3. 一般の場合の計算複雑さ

図 1 のようなシンδροームをもつネットワークを考えてみよう。無矛盾故障集合をあべて列挙すると $\{u_1, u_2, u_3, u_4\}$, $\{u_1, u_3, u_4\}$, $\{u_1, u_2, u_3\}$, $\{u_1, u_2\}$, $\{u_1, u_3\}$, $\{u_3, u_4\}$

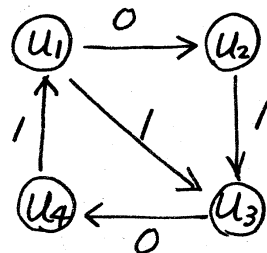


図 1.

となる。通常、故障の発生はユニット数が多くなると同時に起る頻度は減少するという経験的な事実から、真に故障の疑いの強いユニット群として、最小個数の元からなる無矛盾故障集合をその候補として挙げる。これを 最小無矛盾故障集合 と呼び、 $\min. F$ と略す。[†]

[†] 実用的にはさらにユニーク性も仮定されることが多いが、ここでは省く。

一般には $\min. F$ を求める問題を考えるが、おでに、
 “ある正整数 t が与えられたとき、 $|F| \leq t$ なる F が存在
 するか？”という問題が NP -complete となることが示
 されている⁽³⁾⁽⁴⁾⁽⁵⁾。

我々はさらにネットワークの構造として ループ・フリー
 (acyclic) という条件をつけても同じ問題が NP -
 complete となることを指摘しておく。

[定理1] ネットワーク G_N がループ・フリーであっても、
 “ある与えられた正整数 t に対して、 $|F| \leq t$ なる F が
 存在するか？”という問題は NP -complete である。

(証明) 問題は明らかにクラス NP に属す。次に NP -
 complete であることがわかっていて“節点被覆問題”がこ
 の問題に多項式時間で変換できることをいう。

<節点被覆問題> 無向グラフ $G=(V, E)$ に対し、ある
 正整数 t が与えられたとき、 $|X| \leq t$ なる $X \subseteq V$ が
 おべての節点を被覆する (任意の枝の少なくとも一方の節
 点は X に属す) ような X は存在するか？

無向グラフ G に適当に向きをつけてループ・フリーな (acyclic) 有向グラフ G' をつく[†]。次に、 G' のお
 べての有向枝に重み 1 をうべ[†]つ[†]したうべ[†]ル付有向グラフ
 †) 節点に適当なオーダをつけておけば acyclic になる場合。

つ G'' をつくる。この変換が多項式時間でできるのは明らか。さらに、

“節点被覆問題”が解をもつ必要十分条件は、 G が上のようにして G'' に変換されたとき、ネットワーク G'' から $|F| \leq t$ の無矛盾故障集合 F をもつことである。--- も容易にわかる。

q.e.d.

つまり、“向き”に制限をつけても問題の緩和にならないことも定理1は示している。

4. 直並列ネットワークにおける計算複雑さ

ネットワークの構造を直並列に制限すると、最小無矛盾集合を求める多項式時間アルゴリズムが存在する。

[定義] 直並列ネットワーク (s-p ネットワークと略す)

i) 異なる2つの節点 v_i, v_j をもつ枝は2節点 v_i, v_j をもつ s-p ネットワークである。

ii) 2つの s-p ネットワーク E

(a) 図2-(1)のように直列に接続したネットワークは v_i, v_j' を2節点にもつ、

(b) 図2-(2)のように並列に接続したネットワークは v_i, v_j を2節点にもつ、

(それぞれ) S - P ネットワークである。

iii) ii) を有限回くり返して得られるもの (それぞれのみ) S - P ネットワークである。

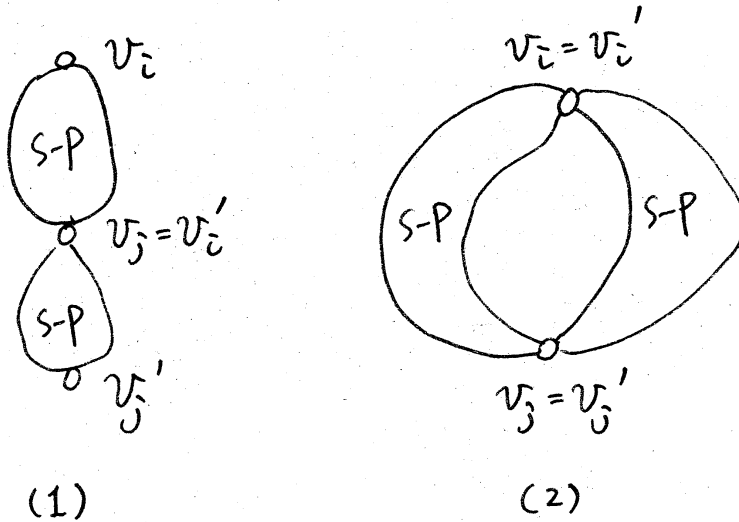


図 2

なお、向きは問題にしていないから、当然、有向ループも存在しうる。

次の定義は証明のためのものである。

[定義] 次の条件を満たす有向グラフ $G=(V, P)$ ($E=E^{\sim}$).

P は $G=(V, E)$ に対し $Px = \{y \mid (x, y) \in E\}$ なる対応をあらわす) E -シーケンシャル・グラフ という。

i) $V = V_0 \cup V_1 \cup \dots \cup V_k$, $V_i \cap V_j = \emptyset$
for any i, j but $i=j$

ii) $V_0 = \{u \mid P^{-1}u = \emptyset\}$, $V_1 = \{u \mid P^{-1}u \subset V_0\}$,

$$V_2 = \{u \mid P^{-1}u \in V_1\}, \dots, V_k = \{u \mid P^{-1}u \in V_{k-1}\}, \\ P^{-1}V_k = \emptyset.$$

シーケンシャル・グラフの例を図3に示す。

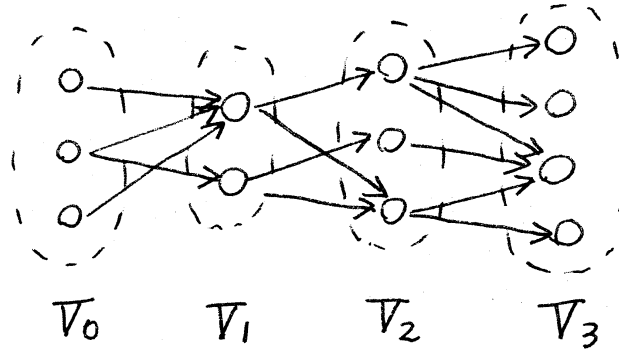


図 3

[補題1] シーケンシャル・グラフにおける最短直内題 (shortest path problem) には $O(|E|)$ のアルゴリズムが存在する。ただし, $|E|$ は枝の数をあらわす。

(証明略)

[定理2] 直並列ネットワークに対して最小無矛盾集合を求める $O(n \log n)$ のアルゴリズムが存在する。ただし, n は節点数である。

(証明) 与えられた直並列ネットワークの2つの節点 (これを u_a, u_b とする) を見出すことは $O(n)$ でできる⁽¹⁾。なお, 一般に直並列ネットワークでは, (2節点間の枝の数が高々1である) 平面グラフの性質 $|E| \leq 3|V| - 6$

が成立つから, $O(|E|)$ と $O(|V|) = O(n)$ は同じである。
 アルゴリズムは "シーケンシャル・グラフにおける最短経路問題" をサブルーチンとして用いる。

すなわち, 図4のようなラベルつきの
 シーケンシャル・グラフを ブロック
ク と呼ぶ。

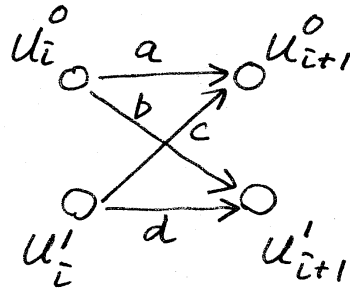


図 4

Procedure A

入力: 図5のようにブロックを $u_{i+1}^0 = u_j^0$, $u_{i+1}^1 = u_j^1$... のように直列につないだラベル付きシーケンシャル・グラフ

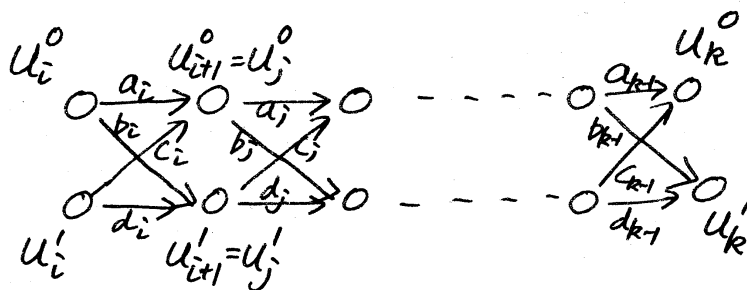


図 5

出力: u_i^x から u_k^y ($x=0,1, y=0,1$) へ至る最短
道の長さ と有向枝のラベルとしてもつブロック

Procedure B

入力: 図6のようなブロックを

$$u_i^0 = u_j^0 = \dots = u_k^0$$

$$u_i^1 = u_j^1 = \dots = u_k^1$$

$$u_{i+1}^0 = u_{j+1}^0 = \dots = u_{k+1}^0$$

$$u_{i+1}^1 = u_{j+1}^1 = \dots = u_{k+1}^1$$

のように並列につないで

シーケンシャル・グラフ

出力: 並列枝のラベルをあべて加

算した値をラベルとして

図7のようなブロック

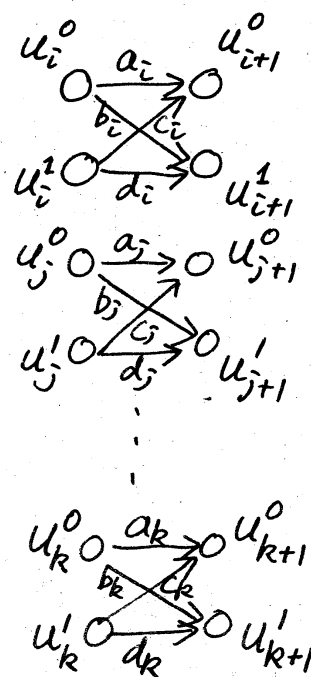


図6

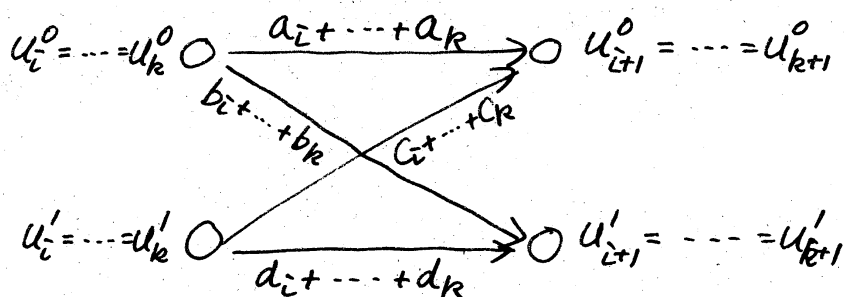


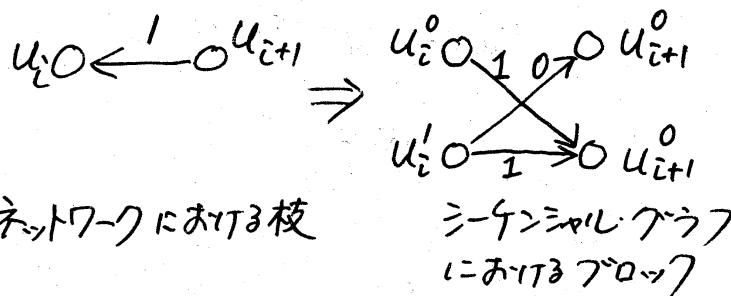
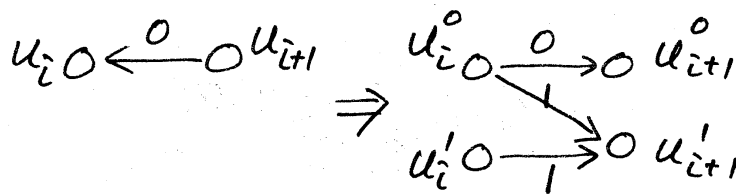
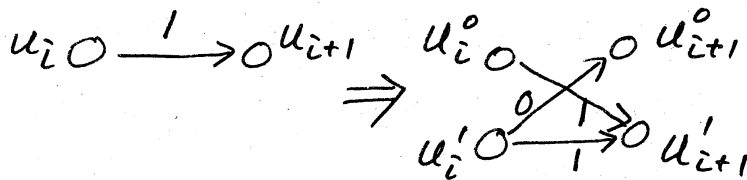
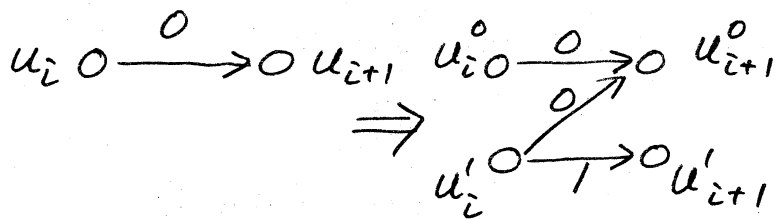
図7

2節点 u_a, u_b をもつ直並列ネットワークに対して

(1) (節点のラベルを残した) 素な道の集合 $\{P_i\}$ に

変換あり。

(2) 素直道 P_i の枝は, ラベルと向きにしたがい, 図8の
ようなブロックに変換される。



ネットワークにおける校

シーケンシャル・グループ
にあるブロック

图 8.

直並列ネットワークの節点 u_a をソースに節点 u_b をシンクとあるフローを考えると各々の素直道 p_i の向きは一意に定まる。その向きに従って図8の変換を行うと p_i から1

つのシーケンシャルグラフが得られる。

(3) それぞれの P_i から対応するシーケンシャル・グラフ G_{P_i} を求める。

(4) それぞれの G_{P_i} に Procedure A を適用する。
(ブロックの個数が減少する。)

(5) ブロックごうしの接続が

直列ならば Procedure A

並列ならば Procedure B

を順次適用して、全体が1つのブロック(図9)になるまで繰り返す。

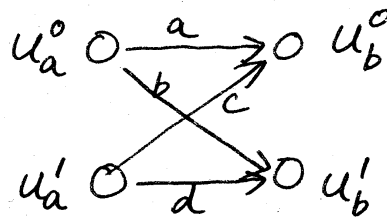


図9

図9において

$a, b, 1+c, 1+d$

のうち最小のものを選ぶ。それが、 $|\min.F|$ であり、

元である故障ユニットの指摘は u_a^x ($x=0,1$) から u_b^y

($y=0,1$) への (シーケンシャル・グラフ上の) 最短道において上添字 1 をとった節点に対応するユニットを集めれば

より。

〈アルゴリズムの正当性〉

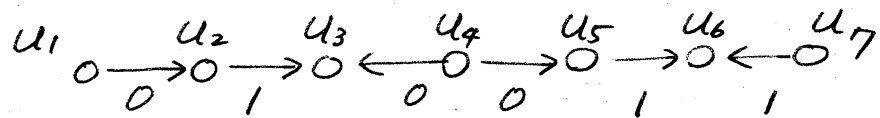
図8の変換はユニットの検査の定義を図示したにほかならない。上添字0はそのユニットが“正常”であることを、1はそのユニットが“故障”であることをあらわす。ブロックにおける有向枝のうべルは“故障ユニット数の増加分”をあらわす。(有向枝が上添字0の節点へ向うときは枝うべルは0, 上添字1の節点へ向うときは枝うべル1となる。)

Procedure A および Procedure B における出力の枝うべルの算出も、ブロックにおける故障ユニット数最小を求めず算法になっていることは容易にわかる。最後のブロック(図9)では u_a' から出る有向枝は u_a 自体が故障の場合であるから1を加えたうべルの値で他と比較する。

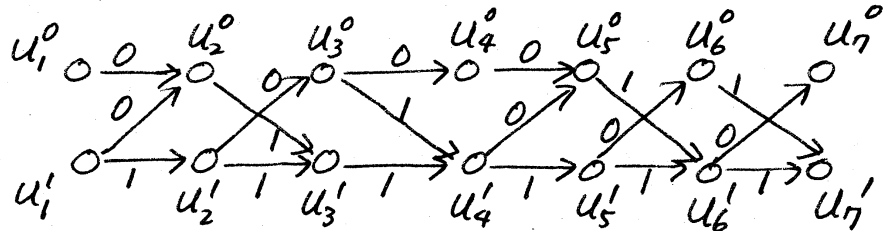
〈計算手数〉

(1) ~ (4) はいずれも高々 $O(n)$ 。 Procedure A も Procedure B もそれぞれは高々 $O(n)$ 。(前者は補題1より, 後者は明らか。) (5) でブロックの縮約のために繰返す手数は高々 $C \log n$ (C は定数)回。したがって, 手数の総計は高々 $O(n \log n)$ である。

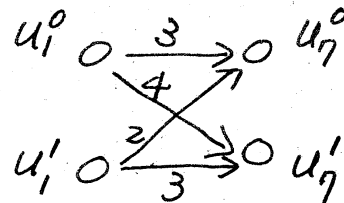
q. e. d.



(a)



(b)



(c)

図 10

簡単のため ステップ (3) と (4) の例だけ図 10 に示しておく。もし、図 (10) (a) が与えられたネットワークであれば (c) で u_1^0 から u_7^0 への枝 (ラベル 3) と u_1' から u_7^0 への枝 (ラベル 2 に 1 を加える) が該当あるから、その道を図 10 (b) から求めればよい。このように定理 2 は $\min. F$ をあわせて求めても手数のオーダーは変わらないことを意味している。

なお、定理 2 の系として、直並列ネットワークの部分ネットワークについても $\min. F$ を求める $O(n \log n)$ アルゴリズムの存在があることが示せる。この結果、直並列ネットワークと若干拡張した D チャート・ネットワーク (D チャート・グラフ⁽⁷⁾ であらわされるネットワークの意味) の最小無矛盾集合を求める問題はやはり多項式時間でよいことがわかる。

5. まとめ

故障の性質や診断の方法を変えると多項式時間アルゴリズムの存在があることはすでに知られていたが⁽³⁾⁽⁵⁾、基本的な最小無矛盾集合を求める問題の場合、ネットワーク構造との関係については、単一ループの場合 $O(n)$ の手数でよいことが知られていたにすぎない⁽⁵⁾。

本稿の結果である“直並列 (あるいは D チャート) ネットワークの最小無矛盾集合は多項式時間で求められる”は平面グラフの部分グラフに対する解法を示している。したがって、ネットワーク構造として“平面グラフ”の場合、計算手数がどうなるかはやはり興味が残る⁽⁸⁾。

文献

- (1) F. P. Preparata et al. : "On the connection assignment problem of diagnosable systems".
IEEE Trans., EC-16, 6, pp. 848-854 (Dec. 1967)
- (2) (モデル議論のまとめとして)
中野ほか : "自己診断可能なシステムに関する研究"
信学技報 R 76-10 (1976-07)
- (3) T. Kameda et al. : "A diagnosing algorithm for networks", Information & Control, 29, pp. 141-148 (1975)
- (4) S. N. Maheshwari et al. : "On models for diagnosable systems and probabilistic fault diagnosis", IEEE Trans., C-25, 3, pp. 228-236 (March 1976)
- (5) 藤原ほか : "システム診断における計算複雑度について", 信学技報 EC 77-8 (1977)
- (6) A. V. Aho et al. : "The Design and Analysis of Computer Algorithms", Addison-Wesley (1974)
- (7) 谷口ほか : "Dチャート拡張プログラムの最適化について — IN関数と求めた $O(N)$ アルゴリズム —", 信学論(D), 57-D, 12, pp. 668-675 (1974-12)

(8) M.R. Garey et al. : "The planar hamiltonian circuit problem is NP-complete",
SIAM J on Computing, Σ , 4, pp. 704-714
(1976)