平面グラフの初等閉路による最小被覆を求めるヒューリスティック・アルゴリズム†

武野純一†† 柿倉正義†††
向殿政男†† 片倉慶孝††††

本論文では，平面グラフの初等閉路による最小被覆を求める一つのヒューリスティック・アルゴリズムを提案している．「初等閉路による最小被覆を求める問題」は，工学的応用に関連が深いが，従来のアルゴリズムは，最適解を与えるかわりに計算量の点で問題があった．本論文では工学的観点より，比較的大規模な実用規模のグラフに対する近似解を与えるためのヒューリスティック・アルゴリズムを開発し，それについて述べている．

本アルゴリズムは，次のようなグラフの性質を利用している．

（1） グラフを被覆する初等閉路の最適解の個数の下限は「$D_{max}/2$⌉である．ここで D_{max} とは与えられたグラフの頂点次数の中で最大のもの．

（2） グラフ上の D_{max} を有するすべての頂点（ただし，D_{max} が偶数であれば，$D_{max}-1$ を有するあらゆる頂点も同時に含む）を通過する初等閉路が一つも存在しなければ，グラフを被覆する初等閉路の数の下限が増加する．

なお，本アルゴリズムは，平面グラフにおいて一つの初等閉路が一連の連結した面に対応することに着目し，解となる初等閉路を発見するために，グラフの面を連結する手法を用いており，その計算量は f をグラフの面の数とすると $O(f^4)$ である．

1. まえがき

グラフ構造を解析する問題の一つである「被覆問題」について，現在までに辺の被覆，頂点の被覆などの幾つかの問題が提案されている[1]．著者らは，先に「最小個数の初等閉路によりグラフを被覆する」という新しい最小被覆問題を提案し，2連結平面グラフを被覆する初等閉路の最小個数，およびそれらの初等閉路を求めるアルゴリズムを開発した[2]．このアルゴリズムは，グラフの全初等閉路の数え上げを基本としているため，演算時間やメモリ容量の点で制約を受け，実際問題としては，面数にして 20 程度のグラフを取り扱うのが限界であった[2]．

本論文では，上述の「初等閉路による最小被覆問題」に対して，比較的大きな実用規模の平面グラフにも実行可能であるようなヒューリスティック・アルゴリズムを提案し，これについて考察する．

† A Heuristic Algorithm for the Minimum Covering Problem of a Planar Graph with Elementary Circuits by JUN-ICHI TAKENO, MASAO MUKAIDONO (Faculty of Engineering, Meiji University), MASAYOSHI KAKIKURA (Automatic Control Division, Electrotechnical Laboratory), and YOSHITAKA KATAKURA (Kokusai Tymshare, Ltd.).

†† 明治大学工学部

††† 電子技術総合研究所

†††† 国際タイムシェア(株)

2. 用語

連結した面とは，面の集合 $\{f_1, f_2, \cdots, f_n\}$ であって，任意の二つの面 f_i, f_j に対して面の連鎖 $f_i=f_{K1}, f_{K2}, \cdots, f_{Kl}, \cdots, f_{Ks}=f_j$（ただし $f_{Kl}\,(l=1, \cdots, s)\in\{f_1, \cdots, f_n\}$）が存在して，連続する二つの面 f_{Kl}, $f_{K(l+1)}$ において，共通な境界辺が存在するものをいう．

Gを平面グラフ，EをGのすべての辺集合，C_i をEの部分集合で初等閉路（自分自身で交わらない閉路）となる辺の集合とする．Eの部分集合の族 $B=\{C_i\}_{i\in N}$（Nは添数集合）において，Bに属するすべての集合の和集合がEに等しいとき，BをグラフGの初等閉路による被覆という．とくに，Bの要素の数が最小なるものを初等閉路による最小被覆という．

最優次数とは，グラフ中の頂点の最大次数 D_{max} が奇数のときは D_{max} をいう．また D_{max} が偶数であるときは，D_{max}，および $D_{max}-1$ を共に最優次数という．

未被覆辺とは，初等閉路により，いまだ被覆されていないグラフ上の辺をいう．

3. 本ヒューリスティックス法の説明

3.1 初等閉路と最優次数の関係

本論文で述べるヒューリスティックス法はグラフの

次のような性質に基づいている．

性質 1．グラフを被覆する初等閉路の最小個数Nは，次式により制限を受ける．

$$\lceil D_{max}/2 \rceil \leq N \leq f. \qquad (1)$$

ここで，D_{max}とはグラフの頂点の次数[3]の中で最大のもの，$\lceil X \rceil$ はXと等しいかより大きい最小の整数，fはグラフの面の個数である．

性質 2．最優次数と等しい次数を持つ頂点をすべて含む初等閉路が存在しなければ，グラフを被覆する初等閉路の個数の下限は $\lceil D_{max}/2 \rceil$ より増加する．

性質 1 について説明する．

一つの初等閉路は，グラフ上の各頂点に接合する辺のうち二本ずつを被覆していく．また同一の頂点を二度は通過しないため一つの頂点に接合する辺の三本以上を被覆することはない．従って，グラフの最大次数 D_{max} を有する頂点に接合する辺をすべて被覆するために，少なくとも $\lceil D_{max}/2 \rceil$ 個の初等閉路が必要となる．これはグラフを被覆する初等閉路の個数の下限を与える．また 2 連結平面グラフの一つの面は，一つの初等閉路に対応する（証明略）ため，グラフを被覆する初等閉路の個数は高々面の個数fで十分である．

性質 2 について説明する．

D_{max} が奇数の場合， D_{max} を有する頂点をすべて通過する初等閉路が発見されたとき，元のグラフでその初等閉路上の辺を開放除去したグラフを考えてみると，そのグラフでは最大次数が元のグラフに比べて減少するため，性質 1 により，初等閉路の数が最小でグラフを被覆するという最適解に近づくことができる．また D_{max} が偶数の場合は，D_{max} を持つ頂点，および $D_{max}-1$ を持つ頂点すべてを通過する初等閉路が発見されればよい．

$D_{max}-1$ の頂点を考慮する理由は D_{max} が偶数のとき，次式が成立するからである．

$$\lceil D_{max}/2 \rceil = \lceil (D_{max}-1)/2 \rceil.$$

すなわち，次数が $D_{max}-1$ の頂点に接合する辺をすべて被覆するためにも，D_{max} の頂点の場合と同様，同数の初等閉路を必要とするからである．

以上により，グラフを最小の数の初等閉路で被覆する問題は，グラフ上の最優次数を有する頂点をできるだけ多く含む初等閉路を発見できるかどうかにより大きく左右されることがわかる．

たとえば図 1(a)の場合を考えてみると（(1)式より下限=2），初等閉路 C_1(実線) は最優次数を持つ頂点 V_5 を残したため，V_5 において次数が減少せず，

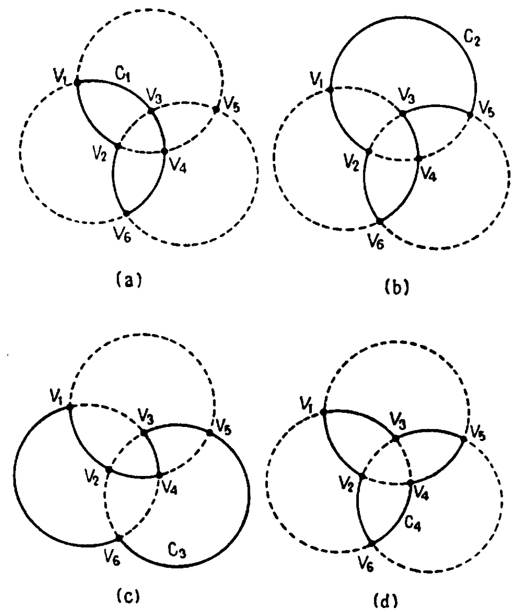

図 1 初等閉路と最優次数を有する頂点との関係
Fig. 1 Relation of an elementary circuit and D_{max} vertices.

性質 1 により V_5 に接合する辺を被覆するために，さらに少なくとも二個の初等閉路が必要となり，初等閉路の数が最小という最適解に近づけない．一方，図 1(b)や(c)のような初等閉路が発見された場合は，グラフの最優次数が減少し，最適解へ近づくことができる．しかし，最優次数を有する頂点をすべて含む初等閉路が，たとえば図 1(d)のようにとられた場合には最適解とはならない．以上により，すべての最優次数の頂点を通る閉路を見いだすという条件は必要条件ではあるが充分条件ではないことがわかる．このゆえに，この性質に基づくグラフの初等閉路による最小被覆の探索法である本アルゴリズムはヒューリスティックスな手法であるといえる．

3.2 解の近似性

本アルゴリズムは，ヒューリスティックス法であるので，得られた解がどの程度最適解に近いか（これを解の近似性と呼ぶことにする）が重要な問題となる．

これには，次のような評価を用いることができる．

与えられたグラフを被覆する初等閉路の最小数，すなわち最適解における個数を N_{opti} とする．グラフの最大次数を D_{max} としたとき，それを被覆する初等閉路の個数の下限 N_{lb} は，(1)式より $N_{lb}=\lceil D_{max}/2 \rceil$ で与えられる．ゆえに一般に次式が成立する．

$$N_{opti} \geq \lceil D_{max}/2 \rceil. \qquad (2)$$

そこで本ヒューリスティックス法で計算された初等

閉路の個数を H_{mini} とすると，解の近似性 P を表わす一つの尺度として，(2)式を用いて次の式を与えることができる.

$$1 \geq P = \frac{N_{opti}}{H_{mini}} \geq \frac{\lceil D_{max}/2 \rceil}{H_{mini}}. \qquad (3)$$

従って，$\lceil D_{max}/2 \rceil = H_{mini}$ であれば，$P=1$ となり，得られた解は最適解である.

4. アルゴリズム概要

本アルゴリズムは，平面グラフにおいて一つの初等閉路が一つの連結した面に対応することに着目し，面を連結する手法を用いて，グラフ中の最優次数の頂点をなるべく多く含み，かつグラフ中の未被覆辺をできるかぎり増加させるような初等閉路を発見していく手法に基づいている.

この考え方を繰り返して用いることにより，グラフの次数を，漸次，最優次数はもちろん，他の頂点の次数をも全体的に下げ，解の収束性および近似性を高めている.

アルゴリズムの概略をピジンアルゴル[4]で表1に記述する.

4.1 アルゴリズムの説明

ALGORITHM 1. MAIN-PROGRAM

アルゴリズムには，グラフ G の各辺に含まれる頂点，各面に含まれる辺および頂点，各面に隣接する面が与えられるものとする.

最初に，未被覆辺の集合 E に，初期値としてグラフ上のすべての辺を与え，さらに，グラフ上のすべての頂点の次数を計算して ORDER へ入れ，次に $E=\phi$ となるまでアルゴリズム 2 の MINIMUMCOVER (E, ORDER) を繰り返し実行する.

ALGORITHM 2. MINIMUMCOVER (E, ORDER)

これは，未被覆辺集合 E を有するグラフにおいて，性質 1，性質 2 に基づいた一つの初等閉路を発見するアルゴリズムである. それには，まず E の最優次数をもつ頂点の集合 MXOD を見つける. 次にそれらの頂点を少なくとも一つは含むグラフの面集合を見つけて INITIALF へ入れる. そして，この INITIALF の各面に対してそれを初期面としてアルゴリズム 3 の CONNECFACE (FACE, CONFACE 0, FACE 1, MARK, C) を用いて解となる初等閉路の候補を計算して集合 C へ入れる. その結果，初期面集合 INITIALF に対応する初等閉路集合 C が発見される. そのとき同

表 1 ヒューリスティック・アルゴリズム

Table 1 Heuristic algorithm.

```
ALGORITHM 1 MAIN-PROGRAM
begin
  E ←— 与えられたグラフのすべての辺集合
  ORDER ←— 全頂点の次数を計算                          ……(1)
  for E≠φ do
    MINIMUMCOVER (E, ORDER)                            ……(2)
end
ALGORITHM 2
procedure MINIMUMCOVER (E, ORDER)
begin
  MXOD ←— E での最優次数を持つすべての頂点の集合        ……(3)
  INITIALF ←— MXOD の頂点を少なくとも一つ含む面の集合
                                                       ……(4)
  C ←— φ
  for 初期面集合 INITIALF の各々の面 f_i に対して do    ……(5)
    begin
      FACE ←— φ
      FACE 1 ←— f_i
      CONFACE 0 ←— φ
      MARK ←— 0
      CONNECFACE (FACE, CONFACE 0, FACE 1,
        MARK, C)                                       ……(6)
    end
  C_o ←— C で最良な初等閉路                            ……(17)
  write (C_o)
  E ←— E-(C_o に含まれる，辺の集合)                    ……(18)
  ORDER の修正
end
ALGORITHM 3
procedure CONNECFACE (FACE, CONFACE 0, FACE 1,
  MARK, C)
begin
  FACE ←— FACE ⊕ FACE 1                               ……(7)
  if FACE が MXOD のすべての頂点を含む then MARK=1
    else MARK=0
  CONFACE 1 ←— (FACE 1 に隣接する面集合 -FACE)
                                                       ……(8)
  TEMP 1 ←— CONFACE 1
  for CONFACE 1 の各々の面 f_i に対して do              ……(9)
    if FACE ⊕ f_i が複数個の閉路となる (検査 A)         ……(10)
      , または非初等化する (検査 B)                     ……(11)
      , または被覆する最優次数の頂点が少なくなる (検査 C)
                                                       ……(12)
    then TEMP 1 ←— (TEMP 1-f_i)
  CONFACE 0 ←— (CONFACE 0-CONFACE 1-FACE 1)
                                                       ……(13)
  for CONFACE 0 の各々の面 f_j に対して do      }
    if FACE ⊕ f_j が非初等化する (検査 D)       }      ……(14)
    then CONFACE 0 ←— (CONFACE 0-f_j)          }
  CONFACE 0 ←— CONFACE 0 ∪ TEMP 1                     ……(15)
  FACE 1 ←— CONFACE 0 の中で最良な面                   ……(16)
  if MARK=1 then if 増加する未被覆辺の数<0
    then FACE 1 ←— φ
  if FACE 1≠φ then CONNECFACE (FACE, CONFACE 0,
    FACE 1, MARK, C)
  else
    begin
      C_k ←— FACE の構成する閉路
      C ←— C ∪ C_k
    end
end
```

時に，その発見された初等閉路集合の各初等閉路に含まれる最優次数の個数および未被覆辺の個数が得られる．そこでこの情報を用いて，得られた初等閉路集合の中より最良な初等閉路を発見する．以後，最良とは，最優次数の頂点をその閉路上に最も多く含むもの，もし最優次数の頂点の個数の同じものがいくつか存在する場合は，その中で未被覆辺数が最も多いものをいう．次に，決定した一つの初等閉路を出力し，未被覆辺の情報Eおよび頂点の次数の情報 ORDER の修正を行う．

ALGORITHM 3. CONNECFACE (FACE, CONFACE 0, FACE 1, MARK, C)

これは再帰的アルゴリズムであって，前回の再帰時における連結途中の初等閉路を構成する面集合 FACE，そのときの FACE に連結可能な連結候補面の集合 CONFACE 0，そのとき CONFACE 0 より実際に選ばれた連結面 FACE 1，スイッチ MARK およびCを伴って実行される（図 2(a)参照）．なお，スイッチ MARK は面集合 FACE によって構成される閉路に最優次数の頂点集合 MXOD の要素がすべて含まれた（MARK=1）か否（MARK=0）かを示す．このアルゴリズムは，面集合 FACE に隣接した面のうち検査 A, B, C, D（これらについては後述する）を通過した連結可能な面の中から最良なものを選んで，FACE に対して MARK のスイッチに従って連結を繰り返すものである．

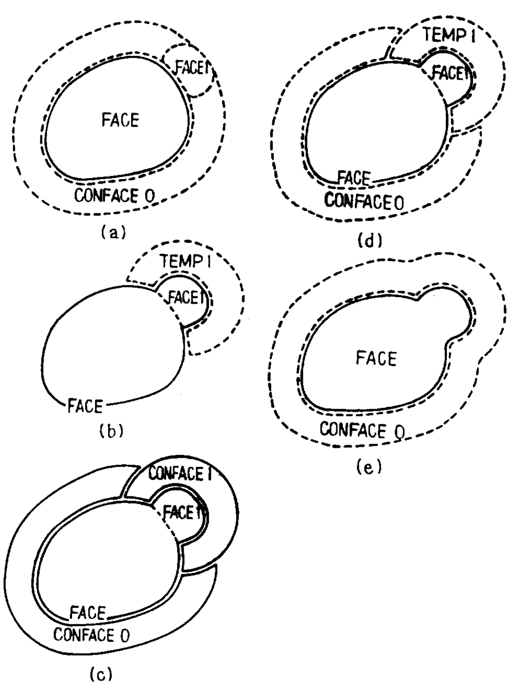

図 2 アルゴリズムの詳細説明のグラフ
Fig. 2 Graphs for explanation.

このアルゴリズムの実行手順は次のとおりである（図2参照）．

まず面集合 FACE に FACE 1 を連結し新たにそれを FACE とする．この新しい FACE に最優次数頂点がすべて含まれたかどうかを検査し MARK に1か0を与える．次に連結面 FACE 1 に隣接する面集合 CONFACE 1 を発見し，さらにその CONFACE 1 より面集合 FACE と共通した面を除去する．次に面集合 CONFACE 1 中に存在するすべての連結候補面の各々に対して，FACE との連結により連結後の閉路が複数になったり（検査A），非初等化したり（検査 B），あるいはいままでの中途初等閉路上に存在した最優次数の頂点を排除したりする（検査C）ことがあるか否かを検査し発見する．CONFACE 1 からこれらの発見された面を除去したものを面集合 TEMP 1 へ入れる（図 2(b)参照）．そのとき同時に，面集合 FACE に TEMP 1 上の各面を連結した場合の最優次数頂点の増加数および未被覆辺の増加数を計算しておく．

CONFACE 0 の各面に対しては前回の再帰時において検査 A, B, C が終了しており，また最優次数頂点の増加数と未被覆辺の増加数についての情報も得られている．現再帰時において FACE 1 が連結されたために，この CONFACE 0 の面の中には，新しく構成された FACE（FACE 1 を含んでいる）に対し連結を行うと，非初等化する場合（検査 D）が存在する．これは CONFACE 0 の面と FACE 1 とが少なくとも一つの頂点を共有する場合である．これを検査するために，面集合 CONFACE 0 より面集合 CONFACE 1 との共通の面を除去し，さらに FACE 1 を除く（図 2(c)参照）．すなわち面集合 CONFACE 0 の各面は，FACE 1 に隣接していない面集合となる．この CONFACE 0 より，検査Dを設けて，非初等化する面を除去する（図 2(d)参照）．

次に CONFACE 0 と TEMP 1 との和集合をとり，それを新しく CONFACE 0 へ入れる（図 2(e)参照）．すなわち CONFACE 0 は，FACE に連結が可能な候補面のすべてとなる．また同時に，CONFACE 0 の各面を FACE に連結することによる，最優次数頂点の増加個数および未被覆辺の増加個数の情報も得られる．よって，これらの情報を用いて，最良の連結候補面を選びそれを FACE 1 とする（図 2

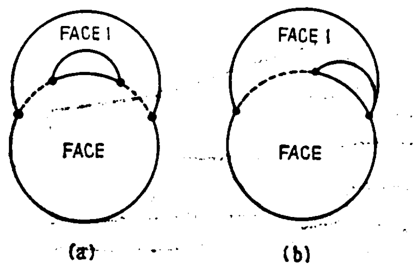

図 3 初等閉路が非初等化する場合の例
Fig. 3 Graphs for explanation.
(a) Separated circuits.
(b) Non-elementary circuit.

(a)参照).

もし，このときスイッチ MARK が1で，かつ未被覆辺の増加数が負となるような場合および連結する面がない場合は，FACE 1 に ϕ が与えられアルゴリズム CONNECFACE を脱出する．それ以外であれば CONNECFACE (FACE, CONFACE 0, FACE 1, MARK, C) へ再帰する．

検査 A，検査 B，検査 C

検査 A を必要とする場合は，面と面との共通部の辺集合が一本の初等的な道[5]を構成しないとき生ずる (図 3(a)参照)．検査 B を必要とする場合は，上記した初等的な道以外の頂点で少なくとも一つの共通な頂点が存在するとき生ずる (図 3(b)参照)．検査 C を必要とする場合は，初等的な道において端点以外の頂点の少なくとも一つが最優次数頂点であるときに生ずる．

5. アルゴリズムの計算量の上限

本ヒューリスティック・アルゴリズムを実行するために，$O(f^4)$ のプログラムを FORTRAN IV を用いて開発した．

アルゴリズムが必要とするデータは，双対グラフを表現する F-F 行列*，F-E 行列**，F-V 行列***，一次元配列対 EV**** である．しかし，実際のプログラム入力では，F-E 行列と EV 配列対のみを与えて，他の行列はこの二つより変換するようにした．

説明は表 1 に従って行う．また説明において e, f, n はグラフの辺数，面数，頂点数を表わす．なお，平面グラフにおいては $O(e)=O(f)=O(n)$ が成立している．

(1)のプログラムはグラフの全頂点の次数を計算するものである．各辺に接合する頂点の情報を持つ一次元配列対 EV のすべての要素を一度だけ探索するので，その計算量は $O(e)$ である．(2)のプログラムの反復回数は，解となる初等閉路の個数であり高々 $O(f)$ である．(3)は，次数情報を持つ一次元配列 ORDER の内容を探索するものであり，その計算量は $O(n)$ である．(4)のプログラムは，一次元配列 MXOD 上の頂点に関して，F-V 行列の各行で論理和を計算し一次元配列 INITIALF へ入れ，次に，その INITIALF 上の各面 f_i の最優次数頂点の個数と未被覆辺の個数とを計算する．ゆえに，その計算量は $O(n\cdot f+e\cdot f)$ である．(5)の反復回数は初期面の個数であり，高々グラフの面数となるから $O(f)$ である．(6)のプログラムの再帰回数は，グラフ上の面を一個ずつ連結するので，これも高々グラフの面数となり $O(f)$ である．(7)および(8)の計算量は $O(e)$ である．(9)の反復回数は，面集合 CONFACE 1 中の要素数であるが，面の隣接数が大域的に最大のグラフ[6]において，その要素数は e, f, n の影響を受けず定数である．(10)は，面集合 FACE の辺と，CONFACE 1 中の一つの面の辺との共通部分を発見する．次に共通の辺上の頂点を求め，その次数を計算するもので，その計算量は $O(e)$ となる．(11)および(12)も同様に計算量は $O(n)$ となる．このとき，検査 A, B, C を通過した連結候補面に対し，FACE と連結することによる最優次数頂点の増加数，未被覆辺の増加数を計算する．この計算量は各各 $O(n), O(e)$ である．

以下同様に(13)～(18)の計算量のオーダーはそれぞれ $O(f)$ であることを示すことができる．

よって開発したプログラムの計算量の上限は，

$$\underset{(1)}{O(f)}+\underset{(2)}{O(f)}\cdot\{\underset{(3)\sim(4)}{O(f^2)}+\underset{(5)\sim(6)}{O(f^2)}\cdot\underset{(7)\sim(16)}{O(f)}+\underset{(17)\sim(18)}{O(f)}\}$$
$$=O(f^4)$$

である．

6. 実行例と検討

計算結果としては，初等閉路を表現する面集合を出

* F-F 行列とは，その要素 a_{ij} を $f_i\in F$ が $f_j\in F$ と隣接すれば 1，そうでなければ 0 としたものである．F はグラフの面の集合である．

** F-E 行列とは，その要素 a_{ij} を $f_i\in F$ が $e_j\in E$ を含めば 1，そうでなければ 0 としたものである．E はグラフの辺の集合である．

*** F-V 行列とは，その要素 a_{ij} を $f_i\in F$ が $v_j\in V$ を含めば 1，そうでなければ 0 としたものである．V はグラフの頂点の集合である．

**** 一次元配列対 EV とは，辺のリストであり，二つの一次元配列を対にして表現される．これら二つの一次元配列 $G=(g_1, g_2, \cdots, g_e)$，$H=(h_1, h_2, \cdots, h_e)$ の各要素は頂点番号である．たとえば「i 番目の辺 e_i は頂点 g_i と頂点 h_i に接合する」と考える．

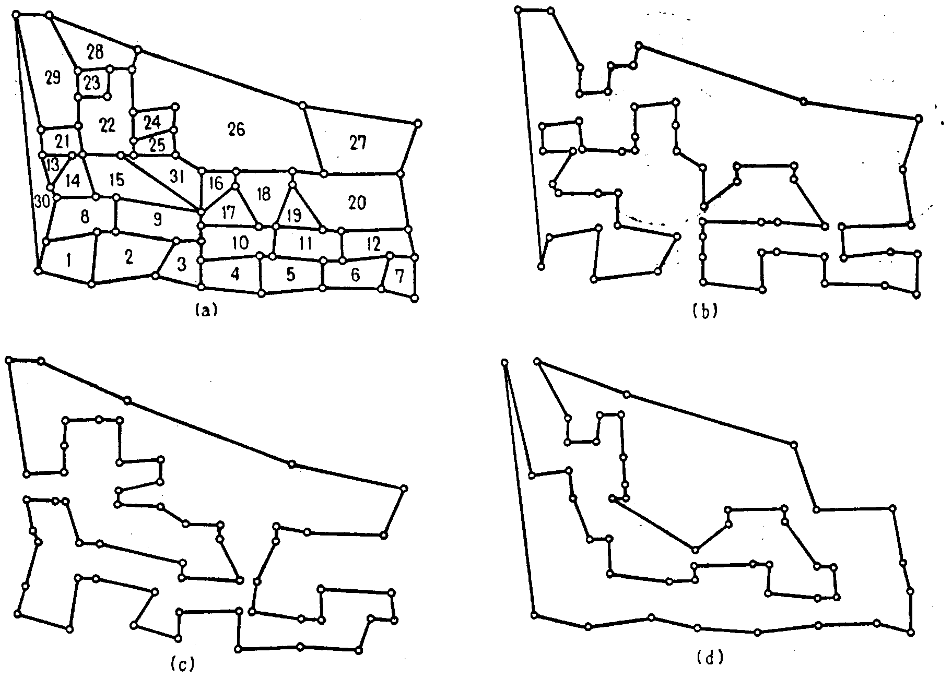

図 4 アルゴリズムの適用
(a) 例題 (b),(c),(d) 解の図形表示
Fig. 4
(a) A graph for an example.
Graph size is 60 vertices, 90 edges, 31 faces.
(b),(c),(d) Schematic representation of a solution.

表 2 例題に対する計算結果
Table 2 Representation of computed result.

RESULT

16 26 22 20 11 10 29 30 8 2 6 4 7 13 27
9 8 10 14 18 26 25 28 29 1 5 6 12 13 3 27
20 26 31 12 6 5 10 3 2 1 4 7 8 14 13 16 21 24 25 28 23 30

力する．

表2は，**図 4**(a) の実行結果であり，図 4(b)，(c)，(d)は解の図示である．このグラフは，60頂点，90辺，31面のある道路網[7]を参考にした2連結グラフである．結果より，このグラフは三つの初等閉路で被覆されることがわかる．この場合，最大次数は5であり，最適解の下限は(2)式より $\lceil 5/2 \rceil = 3$ である．また $H_{\mathrm{mini}}=3$ であるから(3)式を用いると $P=1$ となり，この解は最適解である．

本ヒューリスティック・アルゴリズムをいくつかのグラフに対して適用してみた結果つぎの点が明らかとなった．n_{MXOD} をグラフ中の最優次数頂点の個数，n をグラフ中の全頂点の個数，Av を n_{MXOD}/n とするとき Av が1に近いグラフに対しては，本アルゴリズムの解の近似性が悪化することがある．これは，解が最適解に近づくかどうかの重要なポイントが，グラフの最優次数頂点をすべて含む初等閉路を発見できるかどうかに依存するにもかかわらず，本アルゴリズムはなるべく多くの最優次数を含む初等閉路を発見するようにはたらくため，最優次数頂点をすべて含むことができるという保証がないことによると考えられる．

従って，Av の比較的小さなグラフに対しては，本アルゴリズムが十分に有効であることが示されるが，Av の大きなグラフに対しては他の手法の開発が必要と思われる．

7. むすび

2連結平面グラフの初等閉路による最小被覆問題の解を得る一つのヒューリスティックス法を提案しその有効性を示した．

本ヒューリスティックス法は，グラフ上の最優次数頂点をなるべく多く含む初等閉路の内で，未被覆辺を最も多く含む閉路をその解の候補として選ぶことに基

づいている．

探索技法としては，面の連結に関する breadth-first search[9] を採用している．

本論文で提案したアルゴリズムは，ヒューリスティックスに基づいているため，必ずしも最適解を与えるものではないが，比較的大きな実用規模の平面グラフをとり扱うことができる．

また，開発したプログラムのデータは可能な限りビット・アレイ化して，メモリの減少に努め，かつビット演算を使用し高速化を図っている．しかし，より良いデータ構造を工夫することにより更にプログラムを簡潔にでき，また必要なメモリ量も減少できると思われるが，このことについては現在検討中である．

終りにあたり，日ごろよりご指導をたまわる明治大学講師後藤以紀先生，同教授小川康男先生，並びに電子技術総合研究所佐藤孝平制御部長，長田正情報制御研究室長の各氏に深甚なる謝意を表する．なお，表1の作成に当り電子技術総合研究所塚本享治氏より有益なご助言を得た．ここに記して感謝する．

参考文献

1) 尾崎他：グラフ理論，コロナ社，東京 (1975).
2) 武野，柿倉，向殿，増田：平面グラフの単純閉路による最小被覆を求めるアルゴリズム，情報処理，Vol. 19, No. 11, pp. 1058-1064 (1978).
3) ベルジュ，C.：グラフ理論Ⅰ，サイエンス社，東京 (1976).
4) Aho, hopcroft, Ullman: The Design and Analysis of Computer Algorithms, Addison Wesley, Massachusetts (1975).
5) 野口他：グラフ理論，筑摩書房，東京 (1974).
6) バサッカー，R.G. 他：グラフ理論とネットワーク，基礎と応用，培風館，東京 (1970).
7) Barlow, J.F.: An algorithm for the solution of the postman's problem, Computer Journal, The British Computer Society, London (1978).
8) Christofides, N.: Graph Theory, an Algorithmic Approach, Academic Press, London (1975).
9) Winston, P.H.: Artificial Intelligence, Addison Wesley, Massachusetts (1977).

(昭和53年7月28日受付)
(昭和53年11月1日採録)