数理解析研究所講究録
591巻 1986年 1-11

鎖パッキング問題について

京都大学工学部　増山　繁　MASUYAMA Shigeru
京都大学工学部　茨木俊秀　IBARAKI Toshihide
関西大学工学部　三根　久　MINE Hisashi

あらまし　前稿[8]で明らかにしたように、一般のグラフパッキング問題はＮＰ完全である。そこで前稿では箱も品物も根付き有向木に限り、多項式時間で解ける幾つかの場合を求めた。引続き本稿では、箱に詰める品物を鎖に限った場合に対して考察した。その結果、まず、箱が無向木、品物がｋ鎖(長さｋの道)の場合に最大辺パッキングを求める$O(n \log n)$のアルゴリズム、及び、最大節点パッキングを求める$O(n)$のアルゴリズムを与えた(ｎは節点数)。特に２鎖の場合には最大辺パッキング、最大節点パッキングの何れに対しても、それぞれ$O(n)$のアルゴリズムを得た。一般のグラフに対しては、ｋ鎖の節点パッキング問題は、ｋ＝２でもＮＰ完全であることが知られているが、次数３のグラフに限ってもｋ＝２でＮＰ完全であることを示した。一方、辺パッキング問題に対しては、ｋ＝２の場合には一般のグラフに対しても、与えられたグラフを、まず、深さ優先探索を実行することによって無向木の場合の２鎖の辺パッキングのアルゴリズムを少し修正するだけで、$O(m)$のアルゴリズムを得た(ｍは辺数)。それに対し、$k \geqq 3$のときは、ＮＰ完全であることを明らかにした。

１．はじめに

グラフパッキング問題は、与えられたグラフＧ(箱)にＧの部分グラフ(品物)の集合、G_1, G_2, ･･･, G_tを互いに辺が重ならないように(条件１)、ないし、辺も節点も重ならないように(条件２)詰めることができるかどうかを決定する問題であり、条件１を満たすものを辺パッキング問題、条件２を満たすものを節点パッキング問題という。対応する最適化問題としては、与えられたグラフＧにＧの部分グラフＧ'を条件１(、または２)を満たしつつ最大個数詰める最大辺(節点)パッキング問題や、与えられたグラフＧにＧの部分グラフの集合{G_1, G_2, ･･･, G_t} を条件１ないし２を満たしつつ詰めるのに要する、グラフＧの最小個数を求める問題などがある。

前稿[8]で明らかにしたように、一般のグラフパッキング問題は、ＮＰ完全[1]である。そこで前稿では箱も品物も根付き有向木に限り、多項式時間で解ける幾つかの場合を求めた。本稿では、引続き、グラフに１種類の鎖を最大個数詰める問題に対して得た結果について報告する。なお、以下ではｎはグラフの節点数、ｍはグラフの辺数を表すものとする。

第２節では、箱が無向木、品物がｋ鎖（長さｋの道)の場合に最大辺パッキングを求める$O(n \log n)$のアルゴリズム、及び、最大節点パッキングを求める$O(n)$のアルゴリズムを与える。特に２鎖の場合には最大辺パッキング、最大節点パッキングの何れに対しても、それぞれ$O(n)$のアルゴリズムを得た。

第３節では、一般のグラフへの鎖パッキング問題について考察する。ｋ鎖の節点パッキング問題は、一般のグラフに対してはｋ＝２でもＮＰ完全であることが知られているが、本稿では次数３のグラフに限ってもｋ＝２でＮＰ完全であることを明らかにする。

一方、辺パッキング問題に対しては、$k=2$の場合には、与えられたグラフを、まず、深さ優先探索を実行することによって無向木の場合の2鎖の辺パッキングのアルゴリズムを少し修正するだけで、$O(m)$のアルゴリズムを得た。更に、$k \geqq 3$のとき、NP完全であることを明らかにした。

第4節はむすびで、今後の課題についても言及する。

なお、よく知られたグラフ理論の用語については、定義せずに用いた。詳しくは、文献(1)等を参照されたい。

2．箱が無向木の場合のグラフパッキング

まず、箱が無向木T、品物がk鎖の場合に、辺パッキングを求める$O(n \log n)$のアルゴリズムAを与える。

アルゴリズムA

ステップ1．根から順に幅優先探索を行ない、訪れた順に節点に番号を付ける。$v \leftarrow n$とする。

ステップ2．図1に示すTの部分木T_j内に取りうるk鎖の最大数n_j、e_jを経由してT_j内に取りうる最長路の長さL_jを用いて、図1のT'(節点vを根とするTの部分木)内に取りうるk鎖の最大数n、辺eを経由してT'内に取りうる最長路の長さLを求めよう。 それには、$L_1, L_2, \cdots, L_m$を小さい順にソートし、改めて、$L_1 \leqq L_2 \leqq \cdots \leqq L_m$とする。但し$L_i \leqq k-1$として一般性を失わない。なぜなら、$L_i \geqq k$を満たすものがあれば、予めそこにk鎖を詰めておけばよいからである。

次に、$L_1, L_2, \cdots, L_m$から $L_i + L_p \geqq k$となる組 (L_i, L_p)を最大個数つくる。しかも、残るL_jが最大となるようにする。 (2.1)

そのためには、

$i=1,2,\cdots,k$の順に、以下の操作を続ける。L_iに対し、$L_i + L_p \geqq k$を満たす最小のL_pが見付かるまで二分探索する。組(L_i, L_p)を除去し今の操作を続ける。 (2.2)

この操作を終えた後、

$$n' := \sum_j n_j + (\text{組数})$$

$$L := \text{残った}L_i(i:\max)+1 \qquad (2.3)$$

とする。また、$v \leftarrow v-1$とする。

もし、$v=1$ならステップ3へ、さもなければステップ1へ。

ステップ3．ストップ。n'が木Tに詰めることのできる最大本数のk鎖である。□

<u>定理１．</u>　アルゴリズムＡにより$O(n \log n)$でｋ鎖の無向木Ｔへの最大辺パッキングが求まる。

(証明)手順(2.2)により組を作ることにより、性質(2.1)を満たすことが出来ることを示す。手順(2.2)に従って組を作るときL_1と組になるものをL_pとする。今、性質(2.1)を満たす組合せ方$S=\{(L_1,L_p),\cdots\}$を考える。

<u>場合１．</u>ＳでL_1が使われている場合。

今、$L_{p'}(p'>p)$がＳでL_1と組を作っているとする。このとき、更に次の３つの場合に分れる。

<u>場合１ａ．</u>ＳでL_pが組をつくっていない場合。
このとき、L_1と$L_{p'}$の代りにL_1とL_pとを組にしても性質(2.1)は保たれる。

<u>場合１ｂ．</u>ＳでL_pと$L_k(k<p)$とが組をつくっている場合。
このとき、L_1と$L_{p'}$の代りにL_1とL_pとを、また、L_kとL_pの代りにL_kと$L_{p'}$とを組にしても性質(2.1)は保たれる。

<u>場合１ｃ．</u>ＳでL_pと$L_k(p<k)$とが組をつくっている場合。
このとき、L_1と$L_{p'}$の代りにL_1とL_pとを、また、$L_k+L_{p'}\geqq L_k+L_p\geqq k$であることに注意すると、$L_k$と$L_p$の代りに$L_k$と$L_{p'}$とを組にしても性質(2.1)は保たれる。

<u>場合２．</u>ＳでL_1が使われていない場合。
L_pと組まれているものが無ければL_1をL_pと組む。また、L_pと組まれているものL_jがある場合でもL_1をL_jの代りにL_pと組み直すことができる。同様の議論を手順(2.2)で得られた組(L_i, L_j)に対し、L_iの小さい順に繰返すことにより、手順(2.2)により、性質(2.1)が満たされることが分る。この事実に基づき、アルゴリズムＡにより、ｋ鎖の最大個数の辺パッキングが得られることを、木の高さに関する帰納法で示す。

高さｋまで、アルゴリズムＡで最適解が求まると仮定する。このとき、性質(2.1)により、$T_1, T_2, \cdots, T_m$から、最大個数のｋ鎖が詰められ、かつ、新しい余りＬが最長となることから明らかに高さｋ＋１のときもアルゴリズムＡで最適解が得られる。

さて、アルゴリズムＡの計算時間は、各節点ｉの子の数をm_iとすると、L_i, $i=1, 2, \cdots, m_i$, のソーティングに要する$O(m_i \log m_i)$と、　ソーティングされたL_i, $i=1, 2, \cdots, m_i$,から、操作(2.2)により組をつくるのに要する比較の回数$O(m_i \log m_i)$から決まることから

$$O(\sum_i m_i \log m_i) \leqq O(\sum_i m_i \log n) \leqq O(n \log n) \qquad (2.4)$$

となる。□

特に、ｋ＝２の場合には、以下に示すように$O(n)$で最大個数のｋ鎖の無向木Ｔへの辺パッキングを求めることができる。

<u>アルゴリズムＢ</u>

<u>ステップ１．</u>Ｔに対し、根を適当に決め、根付き有向木とする。まず、根から順に幅優先探索に従って各節点に番号付けする。各節点が存在するかどうかを示す配列$x(i)=1$、for all

iとする(後で節点iが取り除かれると、$x(i)=0$となる)。また、$y=N$, Nはグラフ中の節点数とする。また、各節点に対して、1)子を示すリスト、2)子の数、3)親へのポインター、を、それぞれ記憶しておく。

ステップ2．番号最大の節点vを選び(yを見れば分かる)、節点vに親がないか、もしくは、vの次数が1であり、vの親wの次数も1ならばステップ2へ、さもなければ、vの親wの次数(wの子の数+1)が3以上ならば図2aのように、また、次数が2ならば図2bのように詰める。鎖が詰められた部分はグラフから取り除く。また、それに応じて、y←木に残っている最大の節点番号、とするなど、各データ構造を更新する。ステップ1へ。

ステップ3．ストップ。□

このアルゴリズムで最適解が求まることは、アルゴリズムの実行中にグラフの連結性が保たれ、しかも、最後に、ひとつの節点が残るか、または1本の辺が残るのみであることから明らかである。また、ステップ1で概略述べたデータ構造を用いると、計算時間が$O(n)$で済むことは、容易に分かる。

次に、アルゴリズムAのステップ2を次のステップ2'で置き換えることで、無向木Tへのk鎖の節点パッキングを求めるアルゴリズムCを得る。。

ステップ2'．図3の、$L_1, L_2, \cdots, L_m$のうち、和がk以上になるものを1組だけ見つける。そのためには、最大のものと2番目に大きいものとを組み合わせればよい。

さて、この操作を終えた後、

$$n := \sum_j n_j + (\text{組数}) \qquad (2.5)$$

$$L := \begin{cases} 0 & : v\text{に}k\text{鎖を詰めた場合} \\ \text{残った}L_i(i:\max)+1 & : \text{それ以外の場合} \end{cases}$$

とする。□

定理2．アルゴリズムCにより$O(n)$で木への最大個数のk鎖節点パッキングが求まる。

(証明)任意の最適解を、鎖の詰めなおしの操作を逐次アルゴリズムCで詰められる順に実行していくことにより、アルゴリズムCにより得られた解に修正できることを木の高さに関する帰納法で示す。高さkまでは任意の最適解からアルゴリズムCによる詰め方に修正できると仮定して、高さ$k+1$のときも任意の最適解からアルゴリズムCによる詰め方に修正できることを容易に示すことができる。

さて、今のアルゴリズムの計算時間を評価しよう。アルゴリズムの実行中、手順(2.4)を一回適用するごとに、$L_1, \cdots, L_m$のソーティングに要する$O(m_i \log m_i)$時間かかるだけであり、従って、全体で$O(n \log n)$であることが分かる。□

次に、$k=2$の場合を考える。アルゴリズムは2鎖の辺パッキングの場合のアルゴリズムBでステップ2の最後に「詰めた部分に隣接する辺は全てのぞく」を付加すればよい。これをアルゴリズムDと呼ぶ。これで$O(n)$で最適解が求まることも定理2と同様にして示せる。

3．一般のグラフへのk鎖パッキング

節点パッキングについては、k≧2でNP完全となることが知られている[4]。 更に、グラフの次数を3以下に限っても、NP完全になることを示そう。

定理3．k鎖節点パッキング(k≧2)は次数3以下のグラフに限ってもNP完全である。

(略証)k＝2の場合のみ示す。k≧3の場合も同様にできる。

与えられたグラフGに対して次数4以上の節点を図4に示すように書き換えて、すべての次数が3以下のグラフG'をつくる。すると、

Gにk個以上節点パッキング出来る←ーーー→G'にk＋Σ(d_i－3)以上節点パッキング出来る。

が成り立つことを示すことが出来る。□

系1．2鎖節点パッキングは次数3以下で、しかも、次数3の節点が互に隣接しないグラフに限ってもNP完全である。

(略証)図5に示す帰着により、容易に示すことができる。□

次に、辺パッキングについては、効率よく解ける自明な例として、オイラーグラフがある。オイラーグラフにおけるオイラー道、または、オイラー閉路の長さをmとすれば、k鎖の辺パッキングの最大個数は［m／k］(m／kの整数部分)である。

次に、長さ2の鎖の辺パッキングを求めるO(m)のアルゴリズムを与える。

アルゴリズムE

ステップ1．Gの任意の節点rを選び、根とする。rから、訪れた順に番号を振りながら幅優先探索を行う

ステップ2．vを最大番号の節点とする。もし、vが親をひとつも持たないか若しくはvとその親の次数が共に1のときステップ3へ、さもなければ、図6に示すように2鎖を詰める。□

アルゴリズムEで最適解が求まることは、アルゴリズム実行中に常にグラフの連結性が保たれることに注意すれば、最後に、ひとつの節点が残るか、または一本の辺が残るのみであることから明らかである。また、O(m)時間で実行するためには、アルゴリズムBと同様なデータ構造を用いればよい。

ところが、k≧3とすると、NP完全になることが次の定理4で示せる。

定理4．一般のグラフに対するk鎖辺パッキング問題はk≧3とするとNP完全になる。

(略証)定理3の系1で証明した、次数3以下で、しかも次数3の節点が互いに隣接しないグラフ上での2鎖パッキング問題からの帰着による。まず、与えられたグラフを図7に示すように書き換えて得られるグラフをG'とし、その上での3鎖辺パッキング問題を考えればよい。詳細は紙数の都合上略す。□

表1に鎖パッキング問題の計算複雑度をまとめて掲げる。

4．むすび

今後の課題として、

１．グラフパッキング問題で、今までに得た以外の、効率良く解ける部分クラスを求め、多項式時間で解ける限界を明らかにする、

２．ＮＰ完全である場合に対し、良い近似解法を開発し、その性能評価を行う、

などがある。

謝辞　グラフパッキング問題に対する興味を喚起していただいた、中国、西北電訊工程学院の張澤増副教授に深謝する。

参考文献

(1) Aho, A.V.,Hopcroft, J.E. and Ullman, J.D., The design and Analysis of Computer Algorithms, Addison-Wesley, 1974(邦訳、野崎、野下、共訳、アルゴリズムの設計と解析Ｉ、II、サイエンス社、昭和52年).

(2) Baker, B. S., Coffman, E. G., Jr., and Rivest, R. L., "Orthogonal packings in two dimensions", SIAM J. Compt. 9, pp.846-855, 1980.

(3) Coffman Jr., E. G., Garey, M.R., and Johnson, D. S., "Approximation algorithms for bin packing-an updated survey", working paper, 1984.

(4) Garey, M.,R. and Johnson, D. S., Computers and Intractability:A Guide to the Theory of NP-Completeness, W. H., Freeman and Company, San Fransisco, 1979.

(5) Johnson, D.S., Demers,A., Ullman,J.D., Garey,M.R. and Graham,R.L.,"Worst-case performance bounds for simple one-dimensional packing algorithms", SIAM J. Compt., Vol.3,No.4,pp.601-609,1983.

(6) Kirkpatric, D.G. and Hell, P., "On the complexity of general graph factor problems", SIAM J.Compt.Vol.12, No.3, pp.601-609,1983.

(7) Lawler, E.L., Combinatorial Optimization: Networks and Matroids Holt, Rinehalt and Winston, 1976.

(8) 増山、張、茨木、三根、"グラフパッキング問題の計算複雑度"、理解析研究所考究録556、計算機科学の基礎理論とその応用、pp.260-271、1985.

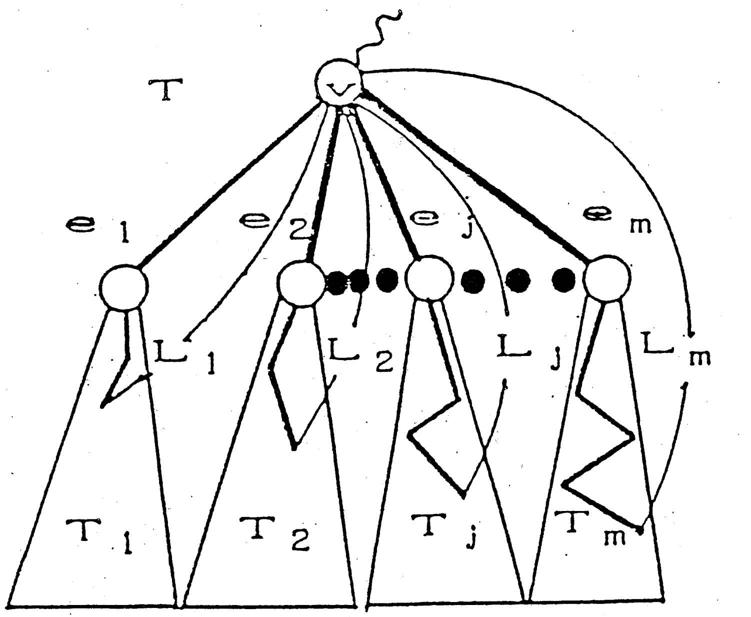
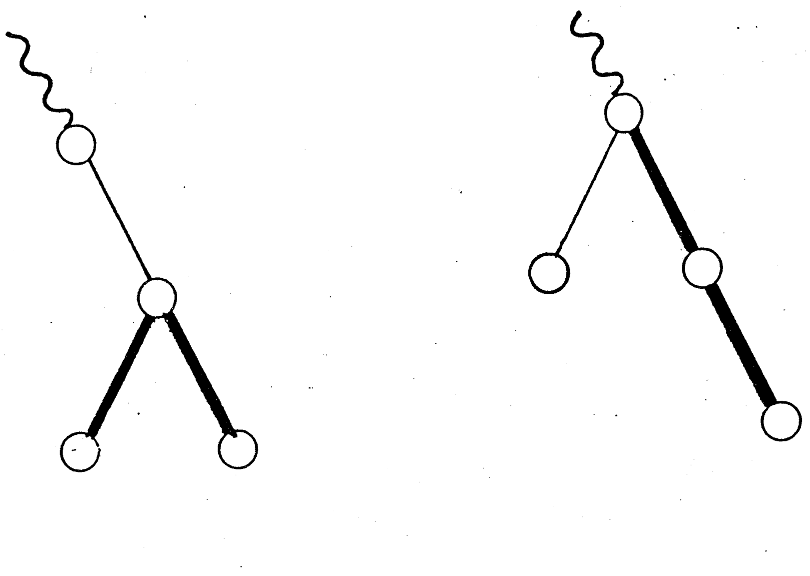


図1．k鎖の木への辺パッキング．



(a)　　　　(b)

図2．2鎖の木への辺パッキング．

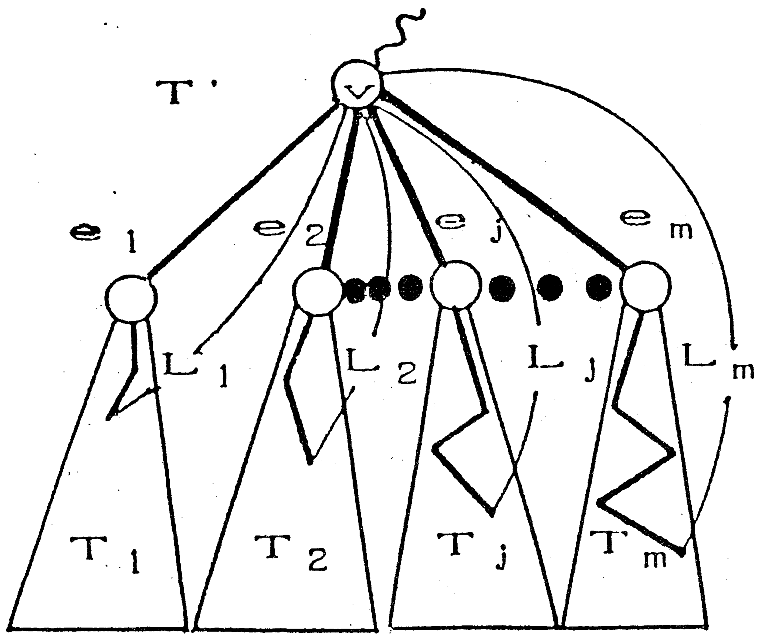


図3．k鎖の木への節点パッキング．

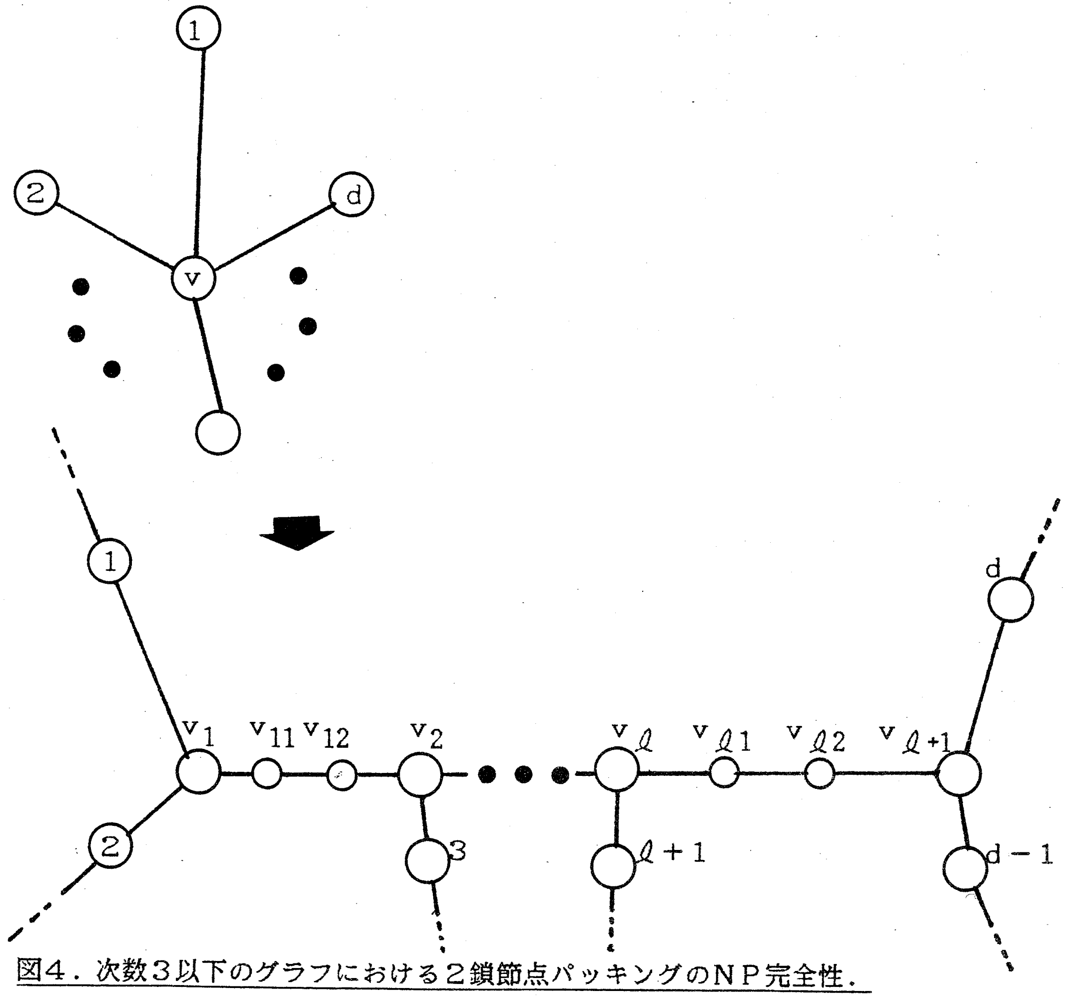


図4．次数３以下のグラフにおける２鎖節点パッキングのＮＰ完全性．

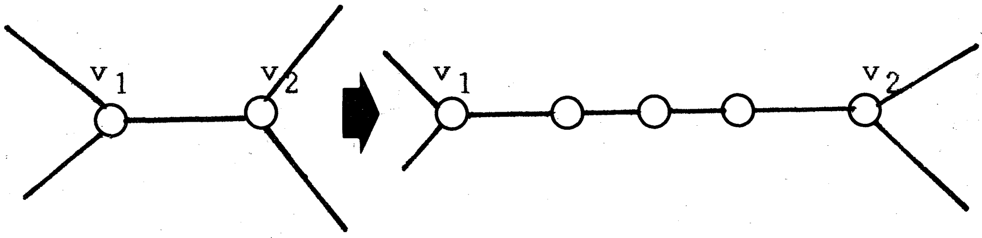


図5．次数３以下でしかも次数３の節点が互いに隣接しないグラフにおける２鎖節点パッキングのＮＰ完全性．

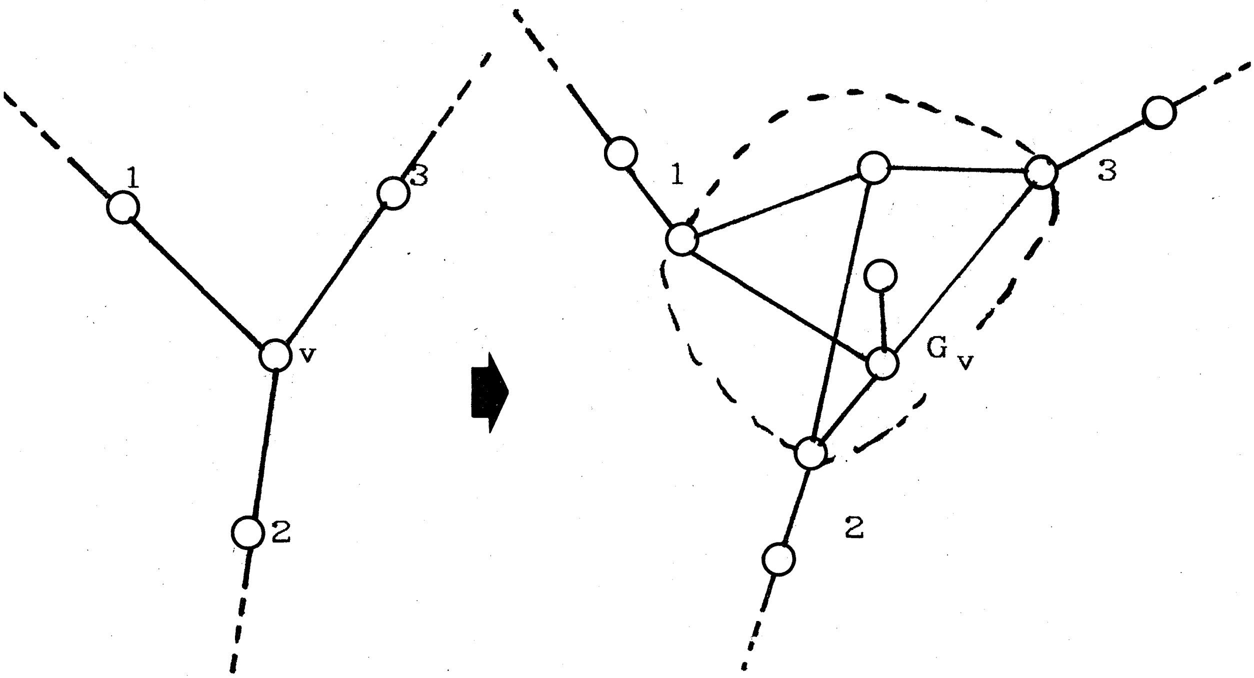
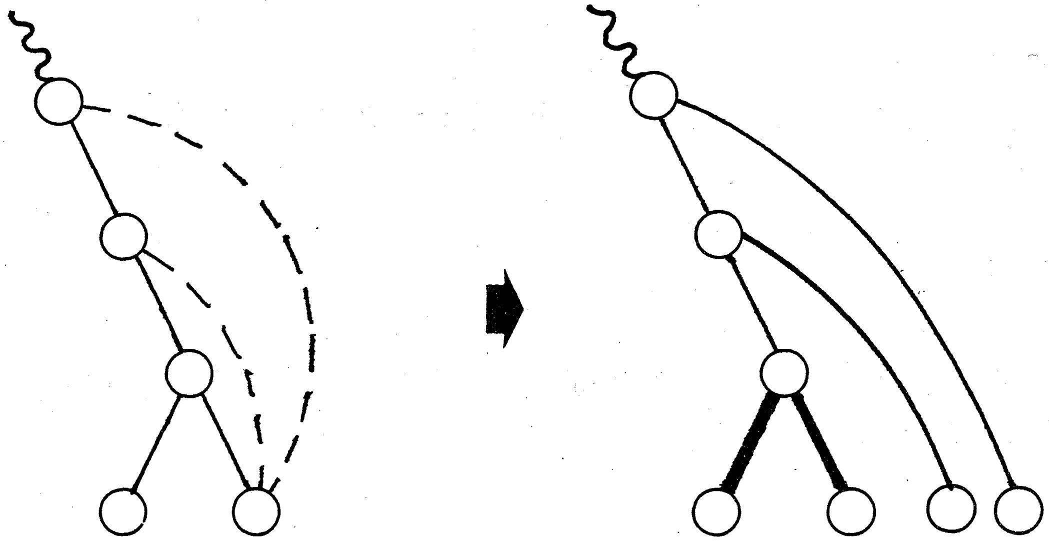
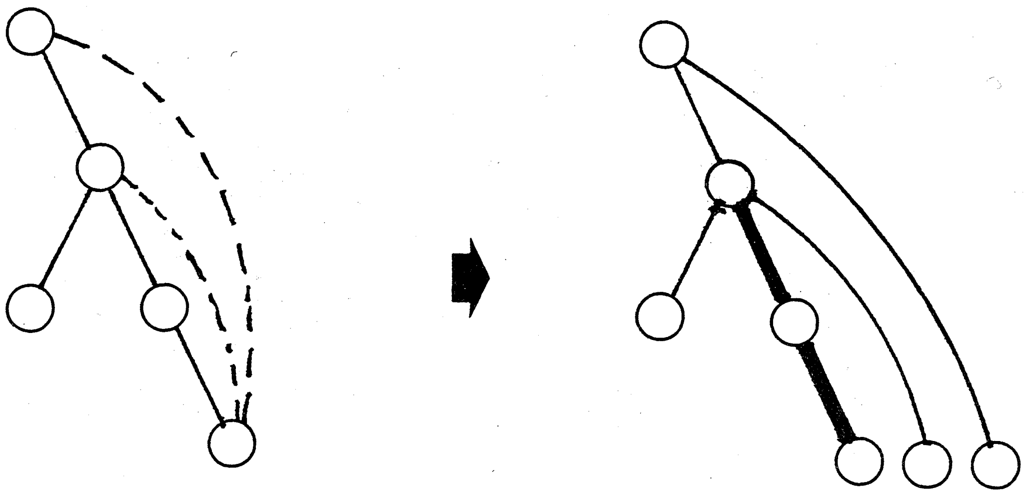


図6．長さ３の鎖の辺パッキング問題のＮＰ完全性(G $_v$は節点ｖに対応する部分グラフを表わす)．



(A)節点ｖに接する少なくとも二つの木の辺があるとき．



(B)節点ｖに接する木の辺が一つしか無いとき．

図７．２鎖の一般のグラフへのパッキング(破線は逆辺を表わす．また、逆辺が無いときは、図２の場合と同様にする)．

箱	辺パッキング	節点パッキング
木 k＝2	O（n）	O（n）
木 k≧3	O（n l o g n）	O（n）
一般のグラフ k＝2	O（m）	NP完全
一般のグラフ k≧3	NP完全	NP完全

ここで、nは箱の辺の数、mは辺の数を表す。

表1．鎖パッキング問題の計算複雑度．