設 計 自 動 化　57－2
（1 9 9 1. 5. 1 0）

モジュール配置問題における分枝限定法の実験的評価

粟島　亨*　　金子　一哉*　　佐藤　政生**　大附　辰夫*

* 早稲田大学理工学部
** 拓殖大学工学部

あらまし

分枝限定法に基づいた２次割当て手法とその計算機実験による評価について述べる．ネットを２点間だけに限定した場合，モジュール配置問題は２次割当て問題として定式化できる．この問題はＮＰ完全問題であり，列挙法の一種である分枝限定法以外に最適解を保証する手法は知られていない．本稿ではまず，厳密な分枝限定法の挙動がその構成法の違いによってどのように変化するかを実験によって評価する．さらに分枝限定法に基づいた近似解法について実験を行った結果，深さ優先探索とγ=0.1の許容誤差法との組合せによって，モジュール数15の問題において近似的最適解が短時間（Sun-4/110上で約5分程度）で得られることが確認できた．

Experimantal Estimation of Branch and Bound Method for Module Placement Problems

Toru Awashima*　Kazuya Kaneko*　Masao Sato**　Tatsuo Ohtsuki*

* School of Science and Engineering, Waseda University
3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169, Japan
** Faculty of Engineering, Takushoku University
815-1 Tatemachi, Hachioji-shi, Tokyo 193, Japan

Abstract

Experimental estimation of the branch and bound tecnique for quadratic assignment problems (QAP) is presented as well as a description of the algorithm. Module placement problems can be formulated as QAP if all nets are restricted to two-terminal nets. QAP is classified into an NP-Complete problem to which only branch and bound method can be applied to obtain an optimal solution. We found that a combination of the depth-first search and the allowance function method where γ is approximately 0.1 is very effective for yielding a suboptimal solution within a reasonable time domain.

1．はじめに

電子回路のレイアウト設計において，複数の機能モジュールを最適な場所（スロット）に配置するというモジュール配置問題は，最も基本的かつ重要な問題の１つである．この問題は，適当な仮定のもとで２次割当て問題として定式化できる[11],[12]．２次割当て問題は，相互間に物流を持つ施設の最適配置問題として提起されたもので，今日まで数多くの手法が提案されてきた[1]~[11]．しかし，提案されている中でも最適解の導出が可能なのは，分枝限定法に基づいた手法だけである[1]~[6]．

分枝限定法は，単なる列挙法に比べれば効率の高い手法ではあるが，多項式時間内での処理を保証するものではない．２次割当て問題の場合，経験的にはモジュール数15程度が許容時間内に解き得る問題規模の限界だと言われている[1]．1976年には，２次割当て問題がＮＰ完全問題であることが証明され[16]，研究の中心は近似解法の開発に移行していった．

分枝限定法の計算コストは決して小さいとは言えない．しかし，問題の規模がそれほど大きくなく，解の最適性がより重要である場合や，多数の解の列挙が必要な場合には効果的である．例えばレイアウト設計の分野では，フロアプランを表現する矩形双対グラフを列挙したり[17]，各論理ブロックのアスペクト比を最適に割り当てることに応用されている[18]．また，階層的分割法との組合せにより，ビルディング・ブロック方式の配置問題に応用した例も報告されている[19],[20]．

分枝限定法に基づいた２次割当て手法については，これまで主に近似解法としての側面から実験的評価がなされてきた[9],[10]．一方，最適解を求める厳密解法については，あまり詳細に評価されたことはなかった．しかし，厳密解法においても，分枝操作や探索法など分枝限定法の基本的構造の違いによって，手法全体の挙動や計算効率がどう変化するのかといったことには興味がある．また，このような性質を明らかにすることは，優れた近似解法を構築する上でも有益であると考えられる．

以下，本稿では２次割当て問題としてのモジュール配置問題を対象とした分枝限定法に焦点を当てて議論を進める．さらに，厳密解法として，あるいは近似解法としての分枝限定法の性質を計算機実験によって評価する．

2．准２次割当てとしての配置問題

本稿では，スロットモデルに基づいたモジュール配置問題を扱う．すなわち，図１のように格子状にn個のスロットが並んでいて，対応するn個のモジュールはこれらのスロット上にのみ配置可能であるというモデルである．各モジュール間に存在する接続要求をネットと呼ぶ．また，各モジュールは少なくとも１つのネットに属しているものとする．通常，目的関数としては総配線長の最小化あるいはこれに適当な重みをつけたものが採用され，このような要請を満たす最適な配置を求めることがモジュール配置問題の目的であると言える．

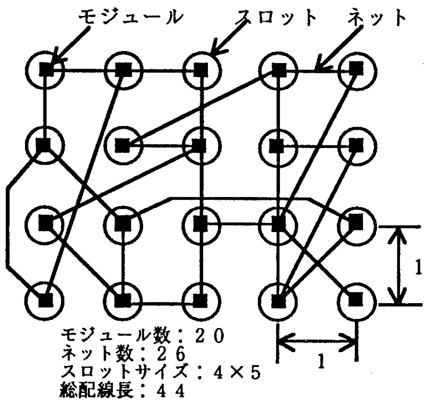

図１　スロット配置モデル

モジュール配置問題は，全てのネットが２点間ネットであるという仮定のもとでは，より一般的な２次割当て問題として定式化できる．また，このような仮定ができないとしても，１組のモジュール対が共通に属するネットの重みの総和をそのモジュール間を接続する２点間ネットの重みとして再定義すれば，２次割当て問題に置き換えることが可能である．このような変換処理により，あらゆるモジュール配置問題を准２次割当て問題として捉えることができる[11],[12]．したがって，２次割当て問題に対する有効な最適化手法について考えをめぐらすことは，モジュール配置問題の立場から見ても有意義であると言える．ただし，このようにして得られた准２次割当て問題の最適解が，必ずしも原問題であるモジュール配置問題の最適解とはならないことに注意が必要である．

以下では，まず２次割当ての特殊な場合として位置づけられる線形割当て問題を定義し，次に２次割当て問題を明確に定義する．

線形割当て問題

この問題は，通常仕事に対して人間を最適に割り当てる問題として定式化できる．いま，n人とnの仕事があって，人iを仕事jに割り当てたときのコストをa_{ij}と表すことにする．また，j=p(i)を人iに割り当てられた仕事とすると，線形割当て問題は，以下のように定式化できる．

$$\text{minimize} \quad F=\sum_i a_{ip(i)} \tag{2.1}$$

線形割当て問題の自明な解法は，n!の可能性を列挙する方法であるが，実際には多項式時間のアルゴリズムが存在する[14]．

２次割当て問題

この問題は，線形割当てのa_{ij}に相当するコスト行列$C=[c_{ij}]$に加えて距離行列$D=[d_{kl}]$を導入したもので，以下のように定式化できる．

$$\text{minimize}\ G=\sum_{i,j} c_{ij} d_{p(i)p(j)} \qquad (2.2)$$

この式のモジュール配置問題における直観的意味は，n個のモジュールとn個のスロットがあって，モジュールiとjの接続の重みがc_{ij}であり，i,jが割り当てられたスロット，p(i),p(j)間の距離が$d_{p(i)p(j)}$で表されているということである．

次節以降では，式(2.2)で定義された２次割当て問題の最適解を求める分枝限定法について基本的概念を整理し，次に２次割当て問題を解く具体的な方法について述べる．なお，今後最適化問題と言ったときは，目的関数の最小化問題のみを指すものとする．またここで定義した２次割当て問題には制約条件が課せられていないので，いかなる解も許容解となる．

３．分枝限定法

最も原始的な組合せ最適化手法は，全ての許容解をくまなく探索することによって解を得る方法であり，列挙法と呼ばれる．組合せ最適化問題においては，ＮＰ完全問題のように，列挙法以外に最適解を求める方法が知られていないものが多数存在する．そこで，極力無駄な探索を抑えながら解の列挙を行おうというのが分枝限定法の基本原理である．分枝限定法は，文字通り分枝操作と限定操作の２つの基本操作から成り立っている．以下では各基本操作について述べる．

分枝操作

原問題を複数の部分問題に分解する操作である．この操作は再帰的に行われ，全ての部分問題が解かれた時点で原問題も解かれたことになる．分枝操作は，具体的には原問題の適当な変数の値を固定することによって行われる．このような変数のことを分枝変数と呼ぶ．問題が分解されていく様子を図２の探索図に示す．探索図とは，実際に生成された部分問題の関係を表したものである．

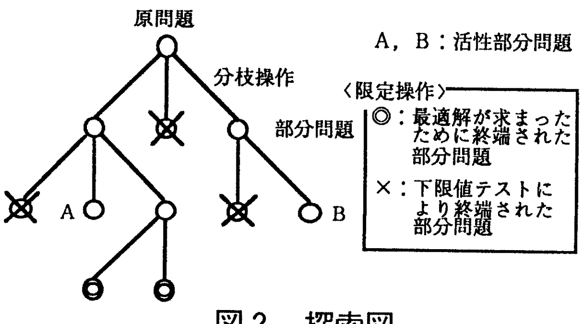

図２　探索図

限定操作

前述の分枝操作だけでは，原始的な列挙法と基本的に同じである．そこで，分枝操作によって生成された部分問題の最適解が直ちに求まる場合や，それ以上分解しても原問題の最適解が得られないことが明らかな場合には，その部分問題を終端し以後列挙の考慮外とする．このような操作を限定操作という．

限定操作は一般に部分問題の下界値によって行われる．いま，現時点までの最良の解を暫定解，その評価値を暫定値とし，分解も終端もされていない部分問題のことを活性部分問題と呼ぶことにする（活性部分問題がなくなったときの暫定解が，原問題の最適解となることは明らかである）．ある部分問題の下界値が暫定値を上回る場合には，それ以上分解しても暫定解を改良することはできないと結論され，その部分問題は終端される．下界値は，通常もとの問題の制約を一部緩めた緩和問題を解くことで計算されるが，ある条件のもとでは，緩和問題の最適解がもとの問題の許容解となり得る．この場合，緩和問題の最適解は同時にもとの問題の最適解であると結論でき，それ以上の分解は必要ないとみなされて，その部分問題は終端される．下界値の善し悪しは，分枝限定法の計算効率に多大な影響をおよぼす．

分枝限定法の挙動を決定するその他の要因の１つとして探索法がある．これは，どのような順序で部分問題を探索していくかを規定したもので，以下の３つが代表的である．

最良下界探索

全ての活性部分問題の中で，下界値の最も小さいものから順に探索する．

深さ優先探索

全ての活性部分問題の中で，探索図における深さの最も深いものから順に探索する．深さが同じ場合には，下界値によって順序を決定する．具体的にはLIFO型のリストによって活性部分問題を保持する．処理のある段階では，リストの末尾の部分問題を取り出してk個の部分問題に分解し，下界値の降順に整列した後再びリストの最後尾に加える．

深さm探索[15]

正整数mを導入することによって一般化された深さ優先探索法である．k個に分解された部分問題をリストの最後尾に加える際に，これらk個の部分問題に加え,末尾のm-1個の部分問題も対象として下界値の降順に整列する．したがって，m=1のとき深さ優先探索と等価となり，m=∞のとき最良下界探索と一致する．

その他に分枝限定法の挙動に影響をおよぼす要因としては，分枝変数の決定法を挙げることができる．

分枝限定法は，適当な処理を付加することで，近似解法として用いることが可能である．最も単純な方法は，計算時間や記憶容量があらかじめ与えた上界に達したとき，処理の全てあるいは一部を強制的に終了させるというもので，打ち切り法と呼ばれる．一方，許容誤差法と呼ばれる近似解法は，分枝限定法の特徴的な処理の１つであり，最適値からの絶対誤差あるいは相対誤差をある一定値内に保って限定操作を効率化する方法である．

次節では，分枝限定法の各構成要素の詳細を２次割当て問題の場合について述べる．ここで，分枝限定法の構造あるいは挙動を表すいくつかの記号を導入する（記号の用法は文献[15]によった）．

P_0　:原問題
P_i　:部分問題
$f(P_i)$　:P_iの最適値
$g(P_i)$　:P_iの下界値
$u(P_i)$　:P_iの上界値
z　:暫定値
T　:計算終了までに分解される部分問題の個数
B　:最適解が暫定解として記憶されるまでに分解される部分問題の個数
F　:最初の許容解が暫定解として記憶されるまでに分解される部分問題の個数
A　:活性部分問題の集合
M　:分枝限定法の計算で得られる｜A｜の最大値

４．分枝限定法による２次割当て手法

本節では前節の内容を踏まえて，２次割当て問題における分枝限定法の各構成要素の詳細について述べる．

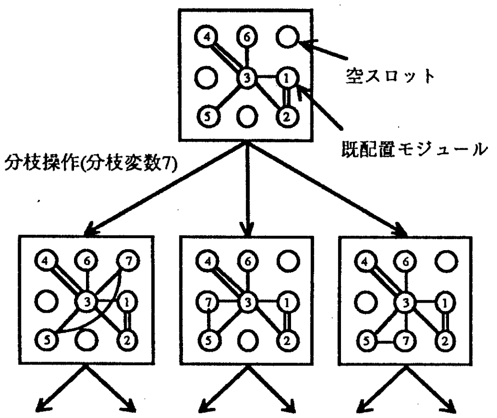

図３　配置問題の探索図

分枝変数

分枝変数としては，モジュール，スロットのどちらを採用することも可能である．ここでは，モジュールを分枝変数として採用することにした．したがって，２次割当て問題としてのモジュール配置問題の探索図は図３のような構造となる．

下界値の計算法

文献[1]の計算法を採用した．これは，あるモジュールをスロットに置いたときに得られる評価値の自明な下界から割当て行列を構成し，線形割当てによって解くことで原問題の下界値を得る方法である（付録Ａ－１参照）．

探索法

探索法としては，より一般性の高い深さm探索法[15]を採用した．実験では，mを1,∞に設定して深さ優先探索，最良下界探索を実現するとともに，m=3,5,10についても評価を行った．一般に最良下界探索はTに関して最良であるが多大なMを必要とし，深さ優先探索はF,BやMに関して有利である．したがって厳密解法には最良下界探索が適しており，近似解法には深さ優先探索が適しているだろうと予想される．その他のmは，mの値に応じて深さ優先探索と最良下界探索の中庸の性質を示すだろうと思われる．

分枝変数の決定法

一般に，Tを改良するためには下界値$g(P_i)$をできるだけ大きくするような分枝変数の決定が有効である．一方FやBを改善したいときは，生成された部分問題の下界値の最小値と最大値の差が最も大きくなるように分枝変数を決定することが効果的である．なぜならば，P_iから生成された部分問題をP_{i1},P_{i2}とした場合，$g(P_{i1}),g(P_{i2})$の値が接近していると，たとえ$g(P_{i1})<g(P_{i2})$であっても，実際には$f(P_{i1})>f(P_{i2})$となる可能性が比較的高いと考えられるからである．つまり，探索すべき部分問題の選定を誤る危険性が高くなるのであって，特に深さ優先探索のとき影響が大きい．また，経験上，より困難なもの，あるいはより重要だと思われるものを分枝変数として選択するという方法もしばしばとられる．実験では，付録Ａ－２に示した７種の分枝変数決定法について検討した．

許容誤差法

許容誤差関数$\varepsilon(z)$を用いて，下界値$g(P_i)$を過大評価する近似解法である．終端条件は次式のようになり，限定操作が効率化される．

$$g(P_i)+\varepsilon(z)\geqq z \qquad (4.1)$$

εがzによらない定数である場合，絶対誤差ε以内の近似最適解が得られる．また，$\varepsilon(z)$が次式，

$$\varepsilon(z)=\gamma\,|z| \qquad (0\leqq\gamma\leqq1) \qquad (4.2)$$

で与えられる場合は，最終暫定値Z_Fは｜Z_F｜に対する相対誤差γ以内の近似最適解である．すなわち，

$$(Z_F - f(P_0)) / |Z_F| \leq \gamma \qquad (4.3)$$

となる．一般に$\varepsilon(z)$の値を大きくすることで，限定操作の効率は上がるが，最初に見つかった許容解以外の解が全く探索されないといったことも起こり得るので注意が必要である．実験では$\gamma=0.001$～0.1程度の相対許容誤差法について評価した．

実験に際しては，以下の点に注目して評価を行う．厳密解法を想定した場合，最適解の導出は保証されているので，計算時間や記憶容量，すなわちTやMの評価が重要である．一方，近似解法においては，TやMに上界値を設けて必要に応じて処理を打ち切る可能性があるため，早い時期に多くの解を求めておくことが重要である．したがって，Tよりも，むしろF,BやMに注目すべきであろう．

5．計算機実験による分枝限定法の評価

実験に用いたプログラムおよびデータの仕様は以下の通りである．

下界値計算：文献[1]の方法（付録A－1参照）
分枝変数　：決定規則1～7（付録A－2参照）
探索法　　：深さm探索法（m=1,3,5,10,∞）
計算機　　：Sun－4／110（C言語）
データ　　：n=6,8,12,15,20,30（付録A－3参照）

全ての実験についてT打ち切り法を適用した．ただし，便宜上部分問題の個数によってではなく，処理に要するCPU時間が1時間を超えたときに無条件に処理を打ち切るものとした．分枝変数の決定規則については，あまり顕著な差異は見られなかったが，問題によらず比較的良好な結果を得た規則2を採用した（付録A－2参照）．なお，実験は最適解の導出を目的とした厳密解法と，近似解法に分けて行った．

5．1 厳密解法に関する実験

n=6,8,12の各データについては，いずれも許容時間内で計算を終了し，最適解を求めることができた．これらのデータに関する結果をもとに，分枝限定法を厳密解法として評価する．

深さm探索法におけるm，モジュール数nとMの関係

図4にモジュール数nと活性部分問題の最大数Mの関係を示す．Mは使用メモリ量と深く関係する．nが大きくなるにしたがってMが増加するのは当然であるが，mの値によって増加の度合いが大きく異なっているのが分かる．m=1の深さ優先探索ではMをnの1次式で抑えることができる．すなわち，

$$M \leq (k-1)n+1 \qquad (5.1)$$

が常に成立する．ただしkは分枝図の各節点が持つ子節点の最大数である．nに対してMが線形であることは，図4からも読みとれる．さらに深さm探索法においては次式が成立することが証明されている．

$$M \leq (k-1)n^m+1 \qquad (5.2)$$

m=∞の最良下界探索では，Mはnに対して指数関数的に増加することが知られている．実験では，n=12の最良下界探索が記憶容量の問題で事実上実行不能であったため断言できないが，m=3～10の結果を見ても，Mのnに対する増加度は，mが大きくなるにつれて飛躍的に増大することは間違いない．さらに，Mの増加度の変化はmが比較的小さいとき大きいと思われた．

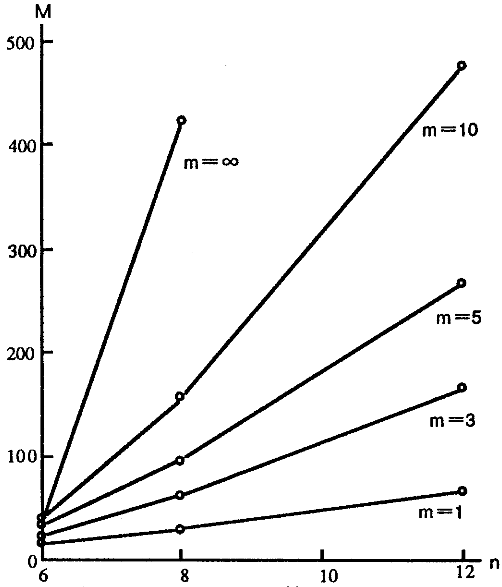

図4　モジュール数nとMの関係

深さm探索法におけるmとT,F,BおよびMの関係

以下ではn=12の結果について述べる．実験は初期暫定値z=∞である一般的な場合と，z<<∞である場合の2通り行った．z<<∞とは，適当な多項式時間のアルゴリズムによって得た解を初期暫定解として与えたことを意味する．ここでは初期解を与えるアルゴリズムとして，分枝限定的考えに基づいた手法であるGilmoreのn^5アルゴリズム[1]を用いた．

初期暫定値z＝∞の場合

図5にmとT,F,B,Mの関係を示す．図より，F,Mを除いた点でmが大きい方が優っており，この場合ではm=10が最良であることがわかる．Tに関する最良下界探索の最良性は理論的にも示されており，実験の結果はこれと合致する．しかし，前述したようにmを大きくするとMすなわち使用メモリ量が飛躍的に増大してしまうため，問題の規模に応じてできるだけ大きなmを設定することが効果的

であると言える.

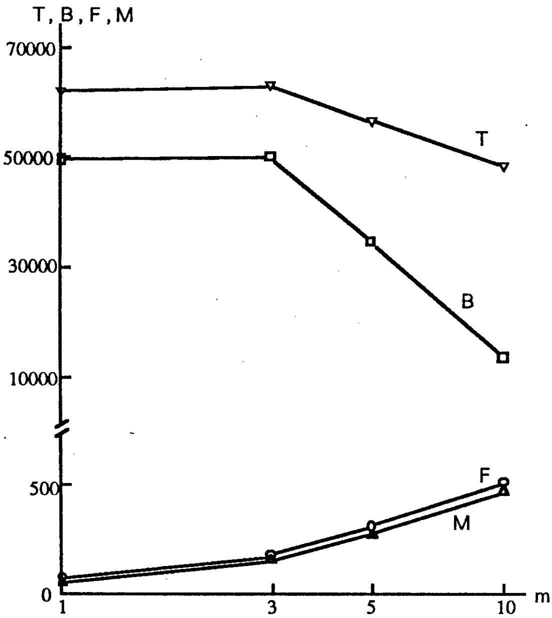

図5　mとT,B,F,Mの関係(z=∞,n=12)

初期暫定値z<<∞の場合

図6にmとT,F,B,Mの関係を示す. 図から明らかなように, T,Bとmの関係は初期暫定値z=∞の場合と完全に逆転している. 最良下界探索においては, Tは下界値関数$g(P_i)$によってほぼ決まり, 上界値はさして意味を持たないと言われている. 初期暫定値は, 探索の初期段階から上界値として機能するが, この効果は深さ優先探索において最も顕著に現れたと言える.

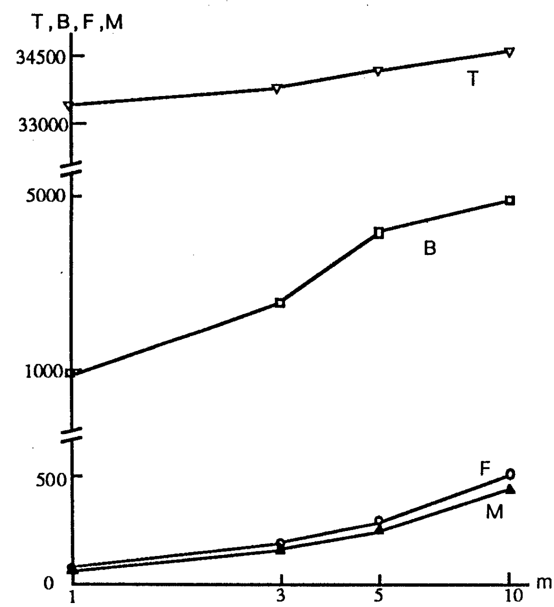

図6　mとT,B,F,Mの関係(z≪∞,n=12)

以上の結果から, 良い初期解を与える方法が存在しない場合には, 記憶容量の許す範囲でできるだけ大きなmを与えた深さm探索が効果的であることがわかった. 一方, 良い初期解を求める多項式アルゴリズムが分かっている場合（本実験はこの場合にあたる）には, これを初期暫定解とした深さ優先探索で十分効果があることがわかった. この場合, 必要な記憶容量を問題規模の線形オーダで抑えられるという利点は重要である.

5. 2 近似解法に関する実験

n=15～30の問題については, 許容時間（CPU時間で1時間）内に最適解を得ることができなかった. そこで, 相対許容誤差法を適用した結果について報告する. 全ての実験は厳密解法に関する結果を受けて, Gilmoreのn^5アルゴリズムによる初期暫定解を与えた深さ優先探索によって行った.

γとTおよび精度μの関係

許容誤差法によって得られた解の精度を見るために, 最適解のわかっているn=12について実験を行った. 図7に式(4.2)におけるγを0.001～0.1に変化させたときのγとTおよび精度μの関係を示す. 精度μは次式で与えられる.

$$\mu=(Z_F-f(P_0))/|f(P_0)| \qquad (5.3)$$

したがってμ=0のとき最終暫定値Z_Fは最適である. γは$|Z_F|$に対する相対誤差の上界であって, μは$|f(P_0)|$に対する相対誤差である. 図より, γを大きくする, すなわちより大きな誤差を許容すると, 生成された部分問題

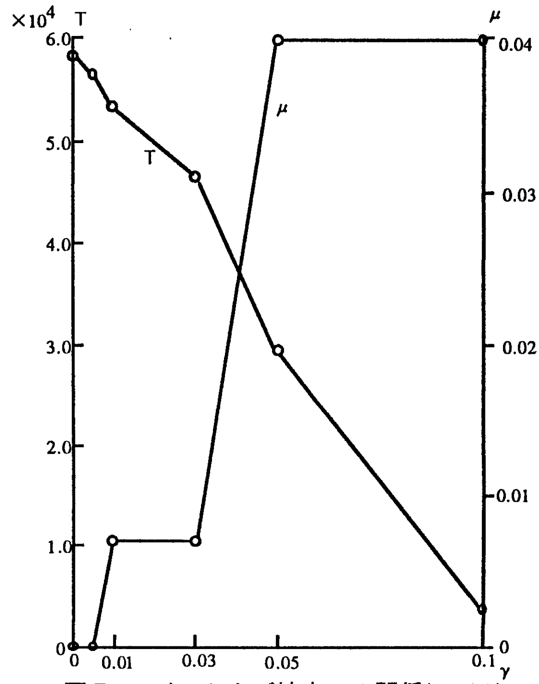

図7　γとTおよび精度μの関係(n=12)

の総数Tは，γの変化にほぼ比例して減少していることがわかる．また，実際に得られた解の精度μはγを常に下回っている．

γと探索済み可能解数の関係

n=15～30のデータについてもγ=0.001～0.1の許容誤差法の実験を行った．結果として，n=15以外のデータでは，許容時間（CPU時間で1時間）内に近似最適解を得ることはできなかった．これらのデータについては，γをさらに大きな値にするか，時間的制約を緩める必要がある．ここではn=15の結果について述べる．

図8にn=15の場合のγとαの関係を示す．ただし，αとは探索の済んだ可能解数を全可能解数を1として表したものである．探索済みの可能解とは終端された部分問題の子孫となる全ての解を含んでいるものとする．図からわかるように，γを大きくするにつれて一定時間（ここではCPU時間で1時間）に探索された可能解数は増加し，γ=0.1のとき全探索を終了している（CPU時間で289秒であった）．このとき得られた近似最適解の評価値は578であった．厳密解法によって最適解を求めたところ（計算には約10時間を要した）その評価値は575であり，非常に精度の高い近似最適解であることがわかる．

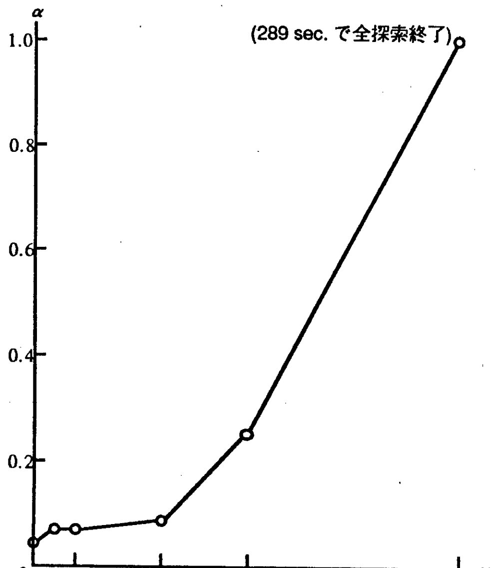

図8　γと探索済可能解αの関係(n=15)
*CPU時間60minで強制的に処理を打ち切った(Sun-4/110)

6．おわりに

本稿では，モジュール配置問題を准2次割当て問題として捉え，分枝限定法によってこの問題を解く手法を実験的に評価した．実験は，最適解を求める厳密解法と近似最適解を求める許容誤差法の両者について行った．いずれの場合もCPU時間で1時間を超える場合は強制的に処理を打ち切るT打ち切り法をとった．

実験の結果厳密解法では，モジュール数n=6～12の問題については許容時間内で最適解が求まった．このとき，比較的良い初期解を与えた場合は，その上界値としての効果により，最良下界探索よりむしろ深さ優先探索の方が効果的であることがわかった．

許容誤差法については，n=15の問題に対してγ=0.1としてやることで許容時間内に精度の高い近似最適解を求めることができた．また，最適解のわかっているn=12の問題にも適用し，近似最適解の精度が通常γによって決まる期待値よりも高いことを確認した．

今後は，まず厳密解法としての立場から手法を洗練する必要があろう．具体的には下界値関数の改良が中心となる．特に准2次割当てとしての近似でなく，一般のモジュール配置問題に直接適用する場合には，解の評価関数も含めた再定義が必要となる．

近似解法としての立場からは，適当なmやγの設定に関する考察が必要であろう．一般にmを小さくすれば，許容解を得るまでの時間や，一定時間内に列挙される許容解の数が増え，使用メモリ量が減少する．γを大きくすると解の最適性は犠牲になるが，限定操作の効率があがる．この場合でも，得られた解の精度を保証できることはこの方法の特徴である．さらに，これらのパラメータを実行中に制御する方法（適応的制御と呼ばれる）についても考えていきたい．

分枝限定法は，多項式時間のアルゴリズムを実現するものではないが，各段階で生成する部分問題を1つに限定すれば，いわゆる組立式の方法となるのであって，そのような手法の一般化と考えることができる．したがってその挙動をうまく制御してやれば，解の最適性や多数の解の列挙といった特徴を持った，柔軟性に富んだ最適化手法と考えることができるのである．

文献

[1]Gilmore, P. C.: Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem, *Journal of the Society Industrial and Applied Mathematics*, Vol. 10, pp. 305-313 (1962).

[2]Lawler, E. L.: The Quadratic Assignment Problem, *Management Science*, Vol.9, pp. 586-599 (1963).

[3]Gavett, J. W. and N. V. Plyter: The Optimal Assignment of Facilities to Locations by Branch and Bound, *Operations Research*, Vol. 14, pp. 210-232 (1966).

[4]Hillier, F. S. and M. M. Connors: Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities, *Management Science*, Vol. 13, pp. 42-57 (1966).

[5]Pierce, J. F. and W. B. Crowston: Tree Search Algorithm for Quadratic Assignment Problems, *Naval Research Logistics Quarterly*, Vol. 18, pp. 1-36 (1971).

[6]Heider, C. H.: An N-Step 2-Variable Search Algorithm for the Component Placement Problem, *Naval Research Logistics Quarterly*, Vol.20, pp. 699-724 (1973).

[7]Steinberg, L.: The Backboard Wiring Problem: A Placement Algorithm, *SIAM Review*, Vol. 3, pp. 37-50 (1961).
[8]Gaschutz, G. K. and J. H. Ahrens: Suboptimal Algorithms for the Quadratic Assignment Problem, *Naval Research Logistics Quarterly*, Vol.15, pp. 49-62 (1968).
[9]Nugent, C. E., T. E. Vollmann and J. Ruml: An Experimental Comparison of Techniques for the Assignment of Facilities to Locations, *Operations Research*, Vol.16,pp.150-173(1968).
[10]Burkard, R. E. and K-H. Stratmann: Numerical Investigations on Quadratic Assignment Problems, *Naval Research Logistics Quarterly*, Vol.25, pp. 129-148 (1978).
[11]Hanan, M. and J. M. Kurtzberg: A Review of the Placement and Quadratic Assignment Problems, *SIAM Review*, Vol. 14, pp. 324-342 (1972).
[12]Breuer, M. A. ed.: *Design Automation of Digital System*, Prentice-Hall (1973).
[13]Preas, B. T. and M. J. Lorenzetti ed.: *Physical Design Automation of VLSI System*, Benjamin/Cummings Publishing (1988).
[14]Burkhard, R. E. and U. Derigs: *Assignment and Matching Problems: Solution Methods with Fortran Programs*, Springer-Verlag (1980).
[15]茨木 俊秀: 組合せ最適化～分枝限定法を中心として～, 講座·数理計画法8, 産業図書 (1983).
[16]Garey, M. R. and D. S. Johnson: *Conputers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, (1978).
[17]Kozminski, K. A. and E. K. Kinnen: Rectangular Dualization and Rectangular Dissections, *IEEE Trans. on Circuit and Systems*, Vol. 35, pp. 1401-1416 (1988).
[18]Wimer, S. et.al.: Optimal Aspect Ratios of Building Blocks in VLSI, *IEEE Trans. on Computer-Aided Design*, Vol. 8, pp. 139-145 (1989).
[19]谷口, 小野寺, 田丸: 配置形状と配線長の制約を考慮したビルディングブロック配置手法, 信学技法, VLD89-114 (1989).
[20]Harkness, C. L. and D. P. Lopresti: VLSI Placement Using Uncertain Costs, *Proc. IEEE Int'l Conf. on Computer-Aided Design*, pp. 340-343 (1990).

付録

A－1　下界値の計算法[1]

いま2つの1次元ベクトル

$$c=(c_1,c_2,\cdots,c_m)$$
$$d=(d_1,d_2,\cdots,d_m)$$

が与えられたとき，cとdの内積が最小となるのは，cの成分が昇順であって，dの成分が降順であるような場合に限られる．この性質を利用して2次割当て問題の下界値を計算することができる．以下に，全てのモジュールが未配置である場合の下界値の計算法を示す．既配置のモジュールが存在する場合でも，既配置モジュールの評価値に対する寄与量を計算して足し合わせれば，ほぼ同様の計算で下界値を求めることができる．

Step1: Cのi行から対角成分c_{ii}を除いた成分によってベクトルc_iを作成する．同様にDのj行からベクトルd_jを作成する．
Step2: c_iとd_jの最小の内積を求め，これをa_{ij}とする割当て行列$A=[a_{ij}]$を作成する．a_{ij}はモジュールiをスロットjに置いた場合の，自明な下界値を表している．
Step3: Aで定義される線形割当て問題を解く．

Step3で求められた線形割当ての最適値が，原問題の下界値となる．

A－2　分枝変数の決定法[1],[4],[10]

実験では7種の分枝変数決定法を比較した．いずれも，A－1で示した割当て行列Aを用いた方法である．分枝変数の決定とは，未配置モジュールの中から適当なモジュールを選ぶことであり，これは割当て行列Aの適当な行を選ぶことに相当する．各行に属する要素の値に注目すると，各決定規則によって選ばれる行（分枝変数＝モジュール）は次のように書ける．

規則1:割当て解の最大の要素が属している行[1]
規則2:最小の要素と最大の要素の差が最大の行
規則3:最小と2番目に小さな要素の差が最大の行[4]
規則4:最大の要素が最大である行
規則5:最小の要素が最大である行
規則6:要素の総和が最大であるような行
規則7:割当て解に属さない最小の要素が最大である行[10]

A－3　実験に使用したテストデータ[9]

実験には，モジュール数n=6,8,12,15,20,30の6種類のテストデータを用いた．これらのデータは文献[9]でNugentらが各種2次割当て手法の比較実験に使ったもので，文献[10]等でもテストデータとして用いられている．モンテカルロ法（試行数10000回）によって解の評価値分布を調べたところ，いずれのデータもほぼ正規分布にしたがっていることを確認した．