囲碁における連数の最大値について

宮 代 隆 平† 矢 野 洋 平†† 村 松 正 和††

囲碁の盤上において，縦または横に連結している同じ色の石の極大集合を連と呼ぶ．連数最大化問題とは，「囲碁のルールの下で n 路盤上に最大でいくつの連が存在できるか」という問題である．この問題はごく最近に提起され，これまでは 16 路盤までの連数の最大値しか求められていなかった．本論文では，連数最大化問題を整数計画問題として定式化し，問題の特徴を利用した制約条件を追加することにより，19 路盤における連数の最大値とその盤面を求めた結果を報告する．

On the Maximum Number of Strings in Go

Ryuhei Miyashiro,† Yohei Yano†† and Masakazu Muramatsu††

In computer Go, a string is defined as a maximal set of connected stones of an identical color. This paper concerns the maximum string problem described as follows: to find a position that maximizes the number of live strings on the $n \times n$ Go board. Previous researches proposed a 0-1 integer programming formulation of the maximum string problem, and solved the instances up to $n = 16$. We reformulate the problem by adding inequalities that break symmetry of the formulation and improve the objective value of linear relaxation. This refinement produces optimal positions up to $n = 19$, the regular size of Go board.

1. はじめに

コンピュータ囲碁において基本的とされている概念に，**連**（string）がある．連とは，「碁盤上において縦または横に連結している同じ色の石の極大集合」である[1)]．**図 1** の 3 路盤には，黒石の連が 2 個，白石の連が 2 個，合計 4 個の連が存在する．連は相手の石によって囲まれた場合に取り上げられる石の集合であるので，囲碁愛好家は思考中にこの概念を使っているはずであるが，あまりに基本的であるためかえって明示的には認識されておらず，「連」という用語も囲碁用語というよりはコンピュータ囲碁用語である．

本論文では，「n 路盤の上に存在しうる連の最大数はいくつか？」という問題を提起する．この問題を n 路盤に対する**連数最大化問題**（Maximum String Problem）と呼び，MSP(n) で表す．混同の恐れがない限り，この最大値自身を MSP(n) と書くこともある．また，最大値を達成するような盤面を最適盤面と呼ぶ．

ここでいう「存在しうる」という部分は，暗黙のうちに「囲碁のルールに適合している」ことを仮定している．囲碁では，ある連に隣接する点がすべて相手の石で埋められたとき，その連は打ち上げられて盤上から取り去られてしまう．したがって，合法な盤面では連に隣接する空点がない状況は許されない．図 1 は合法であるが，**図 2** は合法でない盤面の例である．

連数の最大値 MSP(n) はコンピュータ囲碁のプログラミングにおいて，連情報配列のサイズを与える重要な定数である．この問題はごく最近文献 2) によって初めて提起され，基本的な性質の議論や整数計画問題への定式化が行われた．また整数計画問題を解くことにより MSP(2) から MSP(16) までの値が求められていた．しかし，$n \geq 17$ に対しては同論文で提案されていた手法では計算時間がかかりすぎ，MSP(n) を求めることができていなかった．

一方当然ながら，囲碁としての観点からは MSP(19)，すなわち通常の碁盤上に存在しうる連の最大数に最も興味がある．本論文では文献 2) を発展させ，整数計画問題としての定式化を改良し MSP(19) を求めることに成功した．結論をいえば，MSP(19) = 277 であり，それを達成する盤面の 1 つが図 9 である．

本論文の構成は次のとおりである．まず 2 章で連数最大化問題が呼吸点数最小化問題（後述）に帰着できることを示し，3 章では呼吸点数最小化問題を 0-1 整数計画問題として定式化する．4 章では提案した定

† 東京農工大学
Tokyo University of Agriculture and Technology
†† 電気通信大学
The University of Electro-Communications

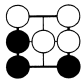

図 1　連数 4 の 3 路盤の盤面
Fig. 1　Four strings on the 3×3 board.

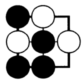

図 2　合法でない盤面
Fig. 2　An illegal position on the 3×3 board.

式化に対し問題の特徴を利用した改善を行うことにより，19 路盤に対する連数最大化問題の最適盤面を求めることができた結果を報告する．

2.　市松模様状の盤面と呼吸点数最小化問題

2.1　市松模様状の盤面

n 路盤の左上隅に白を置き，以下交互に黒石と白石を敷き詰めた盤面を考える．**図 3** は $n = 4$ の場合を示している．この盤面自体は囲碁のルールに従っていないが，ここからいくつか白石と黒石を抜けば，合法な盤面にすることができる．このように，白石と黒石を交互に敷き詰めた盤面からいくつか石を抜き，合法な盤面にしたものを**市松模様状の盤面**ということにする．次の定理は重要である．

定理 1　MSP(n) の最適盤面の中には，市松模様状の盤面が必ず存在する．

証明　矢野・村松[2)]，定理 1 を参照．　□

図 4 に MSP(2)，MSP(3)，MSP(4) の最適盤面のうちの 1 つを示す．連数はそれぞれ 2，6，12 である．この図では $n = 3$，4 の場合が市松模様状の盤面であり，$n = 2$ の場合は市松模様状ではない（$n = 2$ の場合は，たとえば盤面の左上に白石，右上に黒石を置けば，市松模様状の最適盤面になる）．

2.2　呼吸点数最小化問題

n 路盤における石の配置パターンの総数は，合法でない盤面も含めると $3^{n \times n}$ 通りになり，n が大きくなると単純な列挙で MSP(n) を求めるのは事実上不可能である．しかし市松模様状の盤面にのみ限定すれば，交点を定めれば石の色は決まり，各交点に石を置くか置かないかだけを決定すればよいので，パターンの総数は $2^{n \times n}$ となる．この観点から以下，「呼吸点数最小化問題」という問題を定義しよう．

石が置かれていない点を**呼吸点**（liberty）と呼ぶ．白石と黒石を交互に敷き詰めた盤面から石をいくつか取り除いて，（合法な）市松模様状の盤面を作ることを考える．このとき「取り除く石の個数をできるだけ少なくする問題」を**呼吸点数最小化問題**（Minimum Liberty Problem，MLP）と呼び，MLP(n) で表す．また，MSP(n) と同様に，呼吸点数の最小値を MLP(n) で表し，MLP(n) を達成する盤面を最適盤面と呼ぶ．

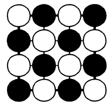

図 3　白石と黒石を交互に敷き詰めた盤面
Fig. 3　The checkered position on the 4×4 board.

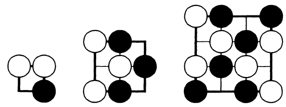

図 4　2〜4 路盤の最適盤面
Fig. 4　Optimal positions for $n = 2, 3, 4$.

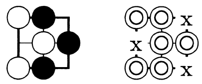

図 5　呼吸点数最小化問題での盤面の表記
Fig. 5　Description of a position in the minimum liberty problem.

定理 2　MLP(n) の最適盤面は MSP(n) の市松模様状の最適盤面であり，逆も成り立つ．さらに，$\mathrm{MLP}(n) + \mathrm{MSP}(n) = n^2$ が成り立つ．

証明　MLP(n) の最適盤面は合法な盤面であるので，そこに置かれている石の個数 s は MSP(n) 以下である．また，MSP(n) の市松模様状の最適盤面における呼吸点の数 l は MLP(n) 以上である．したがって

$$\begin{aligned}\mathrm{MLP}(n) + s &\le \mathrm{MLP}(n) + \mathrm{MSP}(n)\\ &\le l + \mathrm{MSP}(n)\end{aligned}$$

が成り立つが，一方 $\mathrm{MLP}(n) + s = n^2 = \mathrm{MSP}(n) + l$ より上式の不等号はいずれも等号で成り立つ．よって $\mathrm{MLP}(n) = l$，$\mathrm{MSP}(n) = s$ でなければならず，$\mathrm{MLP}(n) + \mathrm{MSP}(n) = n^2$ が成り立つ．

MSP(n) の市松模様状の最適盤面において，呼吸点の数はこの関係式より $n^2 - \mathrm{MSP}(n) = \mathrm{MLP}(n)$ である．すなわちその盤面は MLP(n) の最適盤面でもある．逆も同様に成立する．　□

定理 2 より，連数最大化問題の最適盤面を求めるには，呼吸点数最小化問題を解けば十分であることが分かる．呼吸点数最小化問題は，白石と黒石を交互に敷き詰めた盤面から石を抜いて呼吸点を作る問題であり，石の色を考える必要はない．以降，本論文では連数最大化問題のかわりに呼吸点数最小化問題で議論を進めることにする．また盤面を，**図 5** のように「呼吸

点を x」「石を◎」で表記する．

3. 整数計画問題への定式化

本章では，呼吸点数最小化問題を整数計画問題として定式化する．この整数計画問題は次章での数値実験の基本モデルとなるだけでなく，MSP(n)，MLP(n) の上界，下界を与える（定理 3）．

3.1 整数計画問題とは

整数計画問題とは，「与えられた線形制約式および（一部の）変数につけられた整数条件の下で，線形目的関数の値を最小化/最大化する問題」である．たとえば，すべての変数が 0 または 1 をとる 0-1 整数計画問題は，一般的に以下の式で表される．

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^\top \boldsymbol{x} \\ &\text{subject to} \quad A\boldsymbol{x} \geq \boldsymbol{b}, \\ &\qquad\qquad\quad \boldsymbol{x} \in \{0,1\}^d. \end{aligned}$$

整数計画問題は NP-困難な問題であるが，ある程度の規模の問題までなら分枝限定法を用いて実用的な時間で解くことができる．本論文では誌面の都合上，整数計画問題および整数計画法そのものの解説は行わない．整数計画法については文献 3) などを，現在の計算環境でどのくらいの問題が解けるかは文献 4)，5) などを参照してほしい．近年のハードウェアの進歩，およびそれに優るとも劣らないアルゴリズムの進歩により，数千変数の整数計画問題でも通常の計算機で解けるようになってきている．

なお，解くべき問題を整数計画問題として定式化する際に複数の定式化の方法がある場合には，線形緩和問題の最適値がなるべく大きく☆なるような定式化が「良い」定式化であることを触れておく．

3.2 整数計画問題としての定式化

本節では，呼吸点数最小化問題の 0-1 整数計画問題としての定式化[2)] を説明する．以下，盤上の交点の位置を，盤の左上隅が $(1,1)$ となる座標で表す．つまり，盤の右上隅，左下隅，右下隅はそれぞれ $(1,n)$，$(n,1)$，(n,n) となる．

0-1 変数 $x_{i,j}$ を，座標 (i,j) の交点が呼吸点であるとき 1，呼吸点ではない（石がある）とき 0 をとる変数と定義する．このように定義すると，「呼吸点数の最小化」は目的関数 $\sum_{i=1}^n \sum_{j=1}^n x_{i,j}$ の最小化となる．

また，市松模様状の盤面においては連のサイズはすべて 1 であるから，それぞれの連が 1 つ以上の呼吸点に接しているという条件は，すべての石は 1 つ以上の呼吸点に接しているという条件と同値である．したがって，ある交点 (i,j) を考えると，(i,j) またはその上下左右のどこかに呼吸点がなければならない．この制約条件は，線形不等式で $x_{i,j}+x_{i-1,j}+x_{i+1,j}+x_{i,j-1}+x_{i,j+1} \geq 1$ と表現できる．逆にすべての (i,j) について上の不等式が成り立っていれば，すべての連（石）に関して 1 つは呼吸点があるということなので，その盤面は合法である．

なお，盤の外側は呼吸点ではないため，盤の端や隅では不等式の形が若干異なるが，盤の外部にも変数を定義してやれば統一的に扱える．たとえば左上隅 $(1,1)$ における条件に関しては，盤の外部について $x_{0,1} = x_{1,0} = 0$ という制約式を追加すれば $x_{1,1} + x_{0,1} + x_{2,1} + x_{1,0} + x_{1,2} \geq 1$ のように同じ形の不等式で表せる．

まとめると，呼吸点数最小化問題は以下の 0-1 整数計画問題として記述できる．

$$\text{minimize} \ \sum_{i=1}^n \sum_{j=1}^n x_{i,j}$$

$$\begin{aligned} &\text{subject to} \\ &\quad x_{i,j} + x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} \geq 1 \\ &\qquad (i = 1,2,\ldots,n,\ j = 1,2,\ldots,n), \\ &\quad x_{i,0} = 0 \quad (i = 1,2,\ldots,n), \\ &\quad x_{i,n+1} = 0 \quad (i = 1,2,\ldots,n), \\ &\quad x_{0,j} = 0 \quad (j = 1,2,\ldots,n), \\ &\quad x_{n+1,j} = 0 \quad (j = 1,2,\ldots,n), \\ &\quad x_{i,j} \in \{0,1\} \\ &\qquad (i = 1,2,\ldots,n,\ j = 1,2,\ldots,n). \end{aligned}$$

この定式化により，次のことが分かる．

定理 3 MLP(n)，MSP(n) について，以下が成り立つ．

(1) $\mathrm{MLP}(n) \geq n^2/5$，$\mathrm{MSP}(n) \leq 4n^2/5$.

(2) $\lim_{n\to\infty} \mathrm{MLP}(n)/n^2 = 1/5$，$\lim_{n\to\infty} \mathrm{MSP}(n)/n^2 = 4/5$.

証明 $\mathrm{MSP}(n) + \mathrm{MLP}(n) = n^2$ が成り立つので，MLP(n) の場合だけ証明する．(1) は，整数計画問題の最初の制約から自明である．

(2) では，以下に対応する $n \times n$ の大きさの盤面を考える：

$$x_{i,j} = \begin{cases} 1 & \text{if } 2i + j \equiv 0 \pmod 5) \\ 0 & \text{otherwise.} \end{cases}$$

図 6 は上記に対応する盤面の一部を示したものである．ただし簡略化のため，呼吸点のみを示してある．それぞれの列について，呼吸点の数は $n/5 + O(1)$ 個

☆ 最小化問題の場合であり，最大化問題の場合は逆になる．

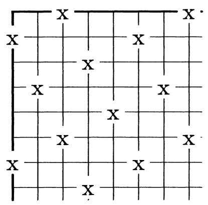

図 6 定理 3 の証明
Fig. 6 Proof of Theorem 3.

表 1 単純な定式化による計算時間
Table 1 Elapsed CPU time with a simple formulation.

n	最小呼吸点数	CPU time (s)
11	29	7.3
12	35	15.5
13	40	53.3
14	47	387.4
15	53	516.3
16	60	1,269.6
17	68	86,461.1
18	76	823,326.1
19	—	—

存在する．この盤面は一般的には合法ではないが，盤端の点すべてを呼吸点に変えると合法な盤面にでき，この操作を行っても呼吸点の数は盤面全体で $O(n)$ 個しか増えない．よって，操作後の呼吸点の総数は $n(n/5 + O(1)) + O(n) = n^2/5 + O(n)$ である．MLP(n) はこの盤面の呼吸点数以下なので，(1) とあわせて $\lim_{n\to\infty} \mathrm{MLP}(n)/n^2 = 1/5$ が成り立つ．

□

4. 整数計画問題を解く

4.1 基礎実験

表 1 は，前章で説明した 0-1 整数計画問題を，商用の整数計画ソルバ ILOG CPLEX 10.1 [6]☆を用いて解いた結果を示したものである．なお計算にはすべて CPU：Pentium 4，2.8 GHz，RAM：3 GB，OS：Windows XP SP2 のマシンを使用した．この表から，n が増加するにつれて爆発的に計算時間が延びていることが分かる☆☆．

3.2 節の定式化では，0-1 変数の数は MLP(n) に対して n^2 個であり，MLP(17) の場合 0-1 変数が 289 個の整数計画問題となる．通常，この程度の規模の整数計画問題は高速に解けることが多いが，MLP(17) では丸 1 日の計算時間がかかり，MLP(18) の計算に 10 日近くかかっている．MLP(19) に関しては計算を打ち切った．

計算の中味を見ると，MLP(17) の最適盤面は計算の開始からわずか 10 分で見つかり，残りの時間は得られた盤面が最適であることの証明に費やされている．整数計画問題において，このように最適性の証明に長く時間がかかるケースというのは，問題に最適解（またはそれに近い値を持つ解）がたくさん存在することが原因の場合が多い．実際に呼吸点数最小化問題をみると，MLP(5) = 7 であるが，呼吸点が 7 つである盤面（最適盤面）は 22 通り，呼吸点が 8 つである盤面は 1,545 通りある．MLP(6) については，最適盤面は 288 通り，最適盤面より呼吸点が 1 つだけ多いような盤面は 20,896 通りも存在する．その結果として，分枝限定法における枝刈りの効果が弱くなり，計算に時間がかかっていると分析される．

これらをふまえ，以下では「定式化に制約式を追加することにより，線形緩和問題の最適値を上昇させる」という形で整数計画問題の計算の高速化を目指す．追加する制約式には 2 種類ある．1 つは盤の隅に関して成立する特殊な条件を考慮したものであり（4.2 節），もう 1 つは最適盤面の対称性を消す目的で盤の中央に着目したものである（4.3 節）．盤の隅に関する不等式は，n が偶数か奇数かに関係しないが，盤の中央に関するものは n の偶奇により形が異なる．これらの制約式を追加した計算機実験の結果は 4.4 節で述べる．

4.2 盤の隅に関する追加制約

ここでは盤の左上隅について説明する．次の性質を利用する．

定理 4 MLP(n) の最適盤面（ただし $n \geq 3$）で，**図 7** (1) の a の部分に少なくとも 1 つは呼吸点が存在するものがある．

証明 背理法で証明する．すべての最適盤面において，a のすべての部分に石が置いてあるものと仮定する．このとき，a の部分を含めた左上 8 交点のパターンは図 7 の (2)–(5) の 4 通りが考えられる．ここで，(2) と (3) を比べると，双方とも合法であり，(3) の方が呼吸点が少ないので (2) は最適盤面になりえない．さらに (3) のパターンに注目すると，左上 8 交点を (3′) というパターンに取り替えても，呼吸点の数は変わらず，しかも盤面の合法性を失わない．したがってこのようにパターンを取り替えた盤面は a の一部に呼吸点のある最適盤面になっており，仮定に矛盾する．同様

☆ $n \geq 17$ の場合では計算時間の 99%以上が最適性の証明に費やされるので，あらかじめすべてのインスタンスに対して探索のパラメータを最適性の証明に重点をおくように設定している．

☆☆ 文献 2) では同じ定式化で MLP(15) の計算に約 1 時間かかっているが，これは計算に用いた整数計画ソルバの性能差によるものである．

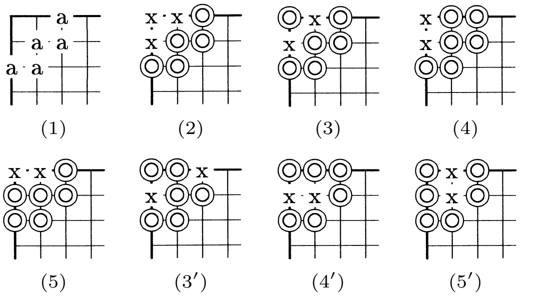

図 7　定理 4 の証明
Fig. 7　Proof of Theorem 4.

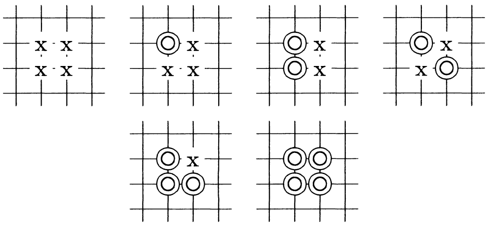

図 8　盤の中央での配置（n が偶数）
Fig. 8　Positions around the center for even n.

に (4) の場合には (4′)，(5) の場合には (5′) というパターンに取り替えればやはりそれぞれが a の一部に呼吸点のある最適盤面になっており，矛盾である．したがって a の部分のどれかが呼吸点であるような最適盤面が存在する．　□

この定理より，$x_{1,3}+x_{2,2}+x_{2,3}+x_{3,1}+x_{3,2} \geq 1$ が成り立つと仮定して一般性を失わない．左上隅だけでなく，他の隅も同様な不等式が導入できる．これらの不等式を追加することにより，0-1 整数計画問題を線形緩和したときの目的関数値が上昇すれば，呼吸点数最小化問題の計算時間が短くなると期待される（実際の効果については 4.4 節で述べる）．

4.3　盤の中央における対称性の除去

最適盤面を左右反転，上下反転，180 度回転させたものも最適盤面である．これらの対称性は，整数計画問題の最適解の個数を増やし，分枝限定法の計算に時間がかかる一因となっている．ここでは盤の中央に注目し，これらの対称性を除去するための不等式を考える．

まず，n が偶数の場合を考慮する．$c=n/2$ とし，盤の中央部の 4 点 (c,c)，$(c{+}1,c)$，$(c,c{+}1)$，$(c{+}1,c{+}1)$ に注目しよう．これら 4 つの交点に石があるか否かのパターンは 16 通り存在するが，対称性を考慮すれば，**図 8** の 6 つのパターンのみを考えれば十分である．したがって，

$$x_{c,c} \leq x_{c,c+1},$$
$$x_{c,c} \leq x_{c+1,c},$$
$$x_{c,c} \leq x_{c+1,c+1},$$
$$x_{c+1,c} \leq x_{c,c+1},$$
$$x_{c+1,c+1} \leq x_{c,c+1}$$

という制約を追加しても一般性を失わない．

n が奇数の場合，$c=(n+1)/2$ とし，盤の中央の交点 (c,c) とそれに接する 4 つの交点 $(c,c-1)$，$(c,c+1)$，$(c-1,c)$，$(c+1,c)$ を考える．これらの 5 つの交点に注目すると，配置の可能性は 32 通り存在するが，n が偶数の場合の議論と同じく，対称性を考慮して次の不等式を追加することができる：

$$x_{c,c-1} \leq x_{c-1,c},$$
$$x_{c,c-1} \leq x_{c,c+1},$$
$$x_{c,c-1} \leq x_{c+1,c},$$
$$x_{c+1,c} \leq x_{c-1,c},$$
$$x_{c,c+1} \leq x_{c-1,c}.$$

これら 5 本の不等式は，n が偶数の場合と本質的に同じであるが，n が奇数の場合は上記 5 つの交点のうち 1 つは必ず呼吸点が存在することに注目すると，さらに不等式 $x_{c,c}+x_{c-1,c} \geq 1$ を追加することができる．これら 6 本の不等式を追加することにより，n が奇数の場合は 32 通りの配置のうち 11 通りのみが許されるようになる．

なお対称性の除去という意味では，別の対称な交点，あるいは対称な領域に注目することもできるが[2)]，整数計画モデルを使う場合，今回の制約式を追加するのが有効なようである．

4.4　計算機実験

本節では，4.2 節，4.3 節で提案した制約式の効果と，それを用いた計算機実験の結果を示す．

表 2　可能な盤面の個数
Table 2　The number of possible positions.

n	呼吸点数	単純	隅	中央	隅＋中央
5	7（最適）	22	22	7	7
5	8	1,545	1,345	571	437
6	10（最適）	288	236	77	64
6	11	20,896	15,760	6,104	4,745

表 2 は，$n=5$，6 の場合に，最適盤面およびそれより呼吸点が 1 個だけ多い盤面がいくつ存在するかを示したものである．項目の「単純」「隅」「中央」「隅＋中央」はそれぞれ「4.1 節の定式化のみ」「4.2 節の不等式を追加」「4.3 節の不等式を追加」「4.2 節，4.3 節の不等式を両方追加」を意味しており，表中の数字が盤面の個数を示している．すでに述べたように，何も不等式を追加しない場合，最適盤面やそれに近いものの個数は非常に多く，計算に時間がかかる要因となっ

表 3 線形緩和問題の最適値
Table 3 Optimal values for LP relaxation problems.

n	単純	隅	中央	隅 + 中央
11	26.8148	26.8148	27.1035	27.1064
12	31.5718	31.6847	31.5718	31.6847
13	36.9291	37.0192	37.0601	37.1144
14	42.4017	42.4927	42.4017	42.4927
15	48.5261	48.6481	48.7067	48.8352
16	54.8129	54.8332	54.8129	54.8332
17	61.7148	61.8148	61.9978	62.0228
18	69.0868	69.0870	69.0868	69.0870
19	76.5302	76.6455	76.7183	76.7916

表 4 呼吸点数最小化問題の計算時間
Table 4 CPU time for the minimum liberty problem.

n	最小呼吸点数	単純な定式化 (s)	制約式を追加 (s)
11	29	7.3	7.3
12	35	15.5	11.4
13	40	53.5	34.0
14	47	387.4	155.2
15	53	516.3	186.9
16	60	1,269.6	1,494.7
17	68	86,461.1	46,765.9
18	76	823,326.1	306,715.2
19	84	—	3,440,752.8

ているが，提案した不等式を加えることにより個数が3分の1〜4分の1程度に減少していることが分かる．盤面の個数を減少させるという意味では，「中央」の不等式が特に有効のようである．

次に，制約式を加えることによる線形緩和問題の最適値の上昇について示す．すでに述べたように，最小化問題を整数計画問題としてモデル化する場合には，線形緩和問題の最適値が大きくなるような定式化ほど望ましい．**表 3** は，それぞれの制約式を加えたうえでの，線形緩和問題の最適値を示している．不等式の追加により，線形緩和問題の最適値が上昇していることが分かる．n が偶数の場合は，n が奇数の場合と比較すると「中央」の不等式を追加する効果が弱いが，これは n が偶数の場合には制約式 $x_{c,c} + x_{c-1,c} \geq 1$ を追加できないことによるものである．

以上で見たように，本論文で新たに提案した制約条件は，「最適盤面（およびそれに近い盤面）の個数の減少」および「線形緩和問題の最適値の上昇」に役立つ．これらの制約条件を追加し，呼吸点数最小化問題を解いた計算結果が**表 4** である．$n \leq 18$ の場合は単純な定式化との比較実験により，例外はあるが不等式の追加による計算時間の減少が確かめられた．単純な定式化と比較すると計算時間が平均で35%減少し，また n が奇数の場合には計算時間がほぼ半分になって

表 5 最小呼吸点数および最大連数
Table 5 The minimum number of liberties and the maximum number of strings.

n	MLP	MSP	n	MLP	MSP
2	2	2	11	29	92
3	3	6	12	35	109
4	4	12	13	40	129
5	7	18	14	47	149
6	10	26	15	53	172
7	12	37	16	60	196
8	16	48	17	68	221
9	20	61	18	76	248
10	24	76	19	84	277

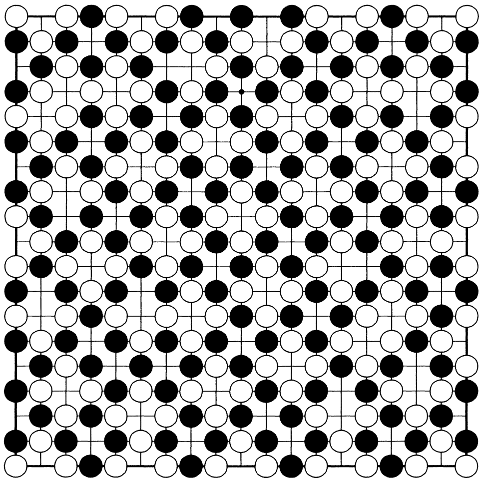

図 9 19 路盤の最適盤面（連数 277，呼吸点数 84）
Fig. 9 An optimal position on the 19 × 19 board.

いる．また MLP(18) の場合は4日弱で，MLP(19) の場合は40日程度☆で計算を終えることができた．最大連数と最小呼吸点数をまとめた表が**表 5** であり，19路盤における最適盤面の1つが**図 9** である．

5. おわりに

本論文では，囲碁の連数最大化問題および呼吸点数最小化問題を考え，これらを整数計画問題として定式化した．また，問題の特徴を生かした制約式を追加することにより19路盤上の最適盤面を求めることができた．整数計画法の観点からは，構造が単純でかつ0-1変数が400個たらずだが，これほど計算時間がかかるインスタンスができることは興味深い．実際の囲碁では，対局中にこのような盤面が出現することはありえないが，コンピュータ囲碁のテスト用盤面として使うことはできるだろう．

☆ MLP(19) の場合も，最適盤面それ自体は計算開始から10分程度で求まっている．

謝辞　本研究の一部は，文部科学省科学研究費補助金若手研究（B）（課題番号 18710127）の助成を受けている．

参　考　文　献

1) 清　愼一，山下　宏，佐々木宣介：コンピュータ囲碁の入門，共立出版 (2005).
2) 矢野洋平，村松正和：碁盤上の連数最大化問題について，第 11 回ゲームプログラミングワークショップ予稿集，pp.107–113 (2006).
3) 今野　浩，鈴木久敏：整数計画法と組合せ最適化，日科技連，東京 (1982).
4) Achterberg, T., Koch, T. and Martin, A.: MIPLIB 2003, *Operations Research Letters*, Vol.30, No.4, pp.361–372 (2006).
5) 宮代隆平，松井知己：ここまで解ける整数計画，システム/制御/情報，Vol.50, No.9, pp.363–368 (2006).
6) ILOG: *ILOG CPLEX 10.1 User's Manual*, Gentilly (2006).

(平成 19 年 1 月 19 日受付)
(平成 19 年 5 月　9 日採録)

宮代　隆平

1999 年東京大学工学部計数工学科卒業．2001 年同大学大学院工学系研究科計数工学専攻修了．2004 年同大学院情報理工学系研究科数理情報学専攻修了．同研究科学術研究支援員，東京農工大学大学院共生科学技術研究部助手を経て，2007 年より同大学大学院共生科学技術研究院助教．博士（情報理工学）．組合せ最適化，数理計画に興味を持つ．日本オペレーションズ・リサーチ学会会員．

矢野　洋平（学生会員）

2006 年電気通信大学電気通信学部情報工学科卒業．現在，同大学大学院電気通信学研究科情報工学専攻在籍．最適化理論，オペレーションズ・リサーチに興味を持つ．

村松　正和

1989 年東京大学工学部計数工学科卒業．1991 年同大学大学院工学系研究科修士課程情報工学専攻修了．1994 年総合研究大学院大学数物科学研究科博士課程統計科学専攻修了．上智大学理工学部助手，電気通信大学情報工学科講師，同学科助教授を経て，2007 年より同学科准教授．博士（学術）．最適化理論，オペレーションズ・リサーチおよびコンピュータ囲碁に興味を持つ．日本オペレーションズ・リサーチ学会，日本応用数理学会各会員．