

# 囲碁における連数最大化問題

宮代 隆平

Ryuhei Miyashiro

東京農工大学

Tokyo University of Agriculture and Technology

矢野 洋平

Yohei Yano

電気通信大学

The University of Electro-Communications

村松 正和

Masakazu Muramatsu

電気通信大学

The University of Electro-Communications

## 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隣接している同じ色の石の極大集合」である。図 1 左の 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。



図 1: 3 路盤. 左は許容解, 右は許容解ではない

本稿では、「 $n$  路盤の上に存在しうる連の最大数はいくつか?」という問題を扱う。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。

ここで言う「存在しうる」という部分は、暗黙のうちに「囲碁のルールに従っている」ことを要請している。囲碁では、ある連に隣接する全ての点が相手の石で埋められたとき、その連は盤上から取り去られてしまう。したがって、隣接する空点が全く無い連が含まれる盤面は許されない。例えば、図 1 左側の盤面は連数最大化問題の許容解であるが、図 1 右側の盤面は許容解ではない。

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。連数最大化問題はごく最近文献 [1] 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なわれた。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。本研究では文献 [1] を発展させ、整数計画問題として

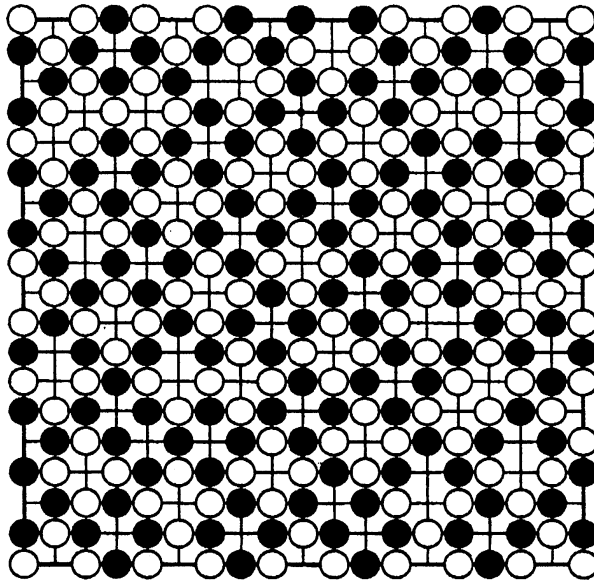


図 2: 19 路盤の最適盤面 (連数 277)

の定式化を改善し MSP(19) を求めることに成功した [2]. 結論を言えば,  $MSP(19) = 277$  であり, それを達成する盤面の一つが図 2 である.

## 2 整数計画問題としての定式化

以下では, 盤上の点の位置を行列と同様に, 盤の左上隅が  $(1, 1)$ , 右上隅が  $(1, n)$ , 左下隅が  $(n, 1)$ , 右下隅が  $(n, n)$  となる座標で表す. 次の定理は重要である.

定理 1 (文献 [1])

連数を最大化する盤面の中に, 以下の条件を満たすものがある:

- $i + j$  が偶数なら, 点  $(i, j)$  は白石か空点である;
- $i + j$  が奇数なら, 点  $(i, j)$  は黒石か空点である.

定理 1 を利用すると, 連数最大化問題は以下の 0-1 整数計画問題として定式化できる [1]. ただし, 0-1 変数  $x_{i,j}$  を, 点  $(i, j)$  が空点のとき 1, 空点ではない (石がある) とき 0 をとる変数と定義する.

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

表 1: 連数最大化問題の計算時間

$n$	2 節の定式化 (s)	3 節の定式化 (s)
11	7.3	7.3
12	15.5	11.4
13	53.5	34.0
14	387.4	155.2
15	516.3	186.9
16	1269.6	1494.7
17	86461.1	46765.9
18	823326.1	306715.2
19	—	3440752.8

### 3 定式化の改善

表 1 の「2 節の定式化」欄は、前節で説明した 0-1 整数計画問題を商用の整数計画ソルバー ILOG CPLEX 10.1 を用いて解いた際の計算時間を表している。  $n$  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。なお計算には全て CPU: Pentium 4, 2.8 GHz, RAM: 3 GB, OS: Windows XP SP2 のマシンを使用した。

前節の定式化では、0-1 変数の数は  $n$  に対して  $n^2$  個であり、MSP(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、MSP(17) では丸 1 日の計算時間がかかり、MSP(18) の計算には 10 日近くかかっている。MSP(19) に関しては計算を打ち切った。

計算の詳細を見ると、MSP(17) の最適盤面は計算の開始からわずか 10 分で見付き、残りの時間は最適性の証明に費やされている。このような計算過程を示すケースは、問題に最適解（またはそれに近い解）が多数存在することが原因の場合が多い。連数最大化問題の場合、 $n$  が小さな値でも最適解が非常に多数存在する。例えば MSP(5) = 18 であるが、定理 1 の条件を満たした上で連が 18 個ある盤面（最適盤面）は 22 通り、連が 17 個ある盤面は 1545 通りある。MSP(6) については、定理 1 の条件を満たす最適盤面は 288 通り、それより連が 1 つだけ少ない盤面は 20896 通りも存在する（表 2 の「2 節の定式化」欄）。その結果、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考えられる。

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、最適解の個数を減らし、また下界値となる線形緩和問題の最適値を上昇させる」という形で計算の高速化を目指す。

まず、盤の隅に注目する。次の定理を利用する。

#### 定理 2 (文献 [2])

連数最大化問題の最適解の中に、定理 1 の条件を満たし、なおかつ図 3 の a の部分に少なくとも 1 つ空点が存在する盤面がある。

表 2: 可能な盤面の個数

$n$	連数	2 節の定式化	3 節の定式化
5	18 (最適)	22	7
5	17	1545	437
6	26 (最適)	288	64
6	25	20896	4745

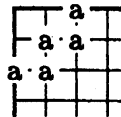


図 3: 空点がある位置

この定理より,  $x_{1,3} + x_{2,2} + x_{2,3} + x_{3,1} + x_{3,2} \geq 1$  という制約式が追加でき, また左上隅だけでなく他の隅も同様な不等式を導入できる.

次に, 盤面の中央に注目し, 対称性を除去することを考える.

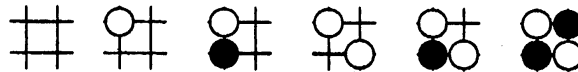


図 4: 盤の中央での配置 ( $n$  が偶数)

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 つの点に石があるか否かのパターンは 16 通り存在するが, 定理 1 および盤面の対称性を考慮すれば, 図 4 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,  $n$  が偶数の場合と同じく, 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.

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 [1], 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.

これらの制約条件を追加することにより, 最適解の個数が減少した様子が表 2 の「3 節の定式化」欄である.  $n = 5, 6$  の場合, 解の個数が約 4 分の 1 に減っている. また本稿では割愛するが, 不等式の追加により線形緩和問題の最適値も上昇することが確かめられた.

不等式を追加して連数最大化問題を解いた際の計算時間を, 表 1 の「3 節の定式化」欄に示す.  $11 \leq n \leq 18$  の場合, 計算時間が平均で 35% 減少し, 特に  $n$  が奇数の場合には計算時間がほぼ半分になっている. また約 40 日の計算により, MSP(19) を解くことができた. (MSP(19) の場合も, 最適解それ自体は計算開始から 10 分程度で求まっている.)

## 4 おわりに

本論文では、囲碁の連数最大化問題を扱い、問題の特徴を生かした制約式を追加することにより 19 路盤の最適解を求めることに成功した。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 個足らずだが、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。最後に、現在あるほとんど全ての囲碁ソフトは、図 2 の盤面を読み込ませると強制終了することを述べておく。

## 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] 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, 107-113, 2006.
- [2] 宮代隆平, 矢野洋平, 村松正和: 囲碁における連数の最大値について. 情報処理学会論文誌, 48 (2007), 3463-3469.