囲碁における連の最大数についての一考

蛭田雄一[†1] 矢野洋平[†2] 五十嵐 力[†3]
但馬康宏[†4] 小谷善行[†4]

囲碁盤における石のまとまりである連をより多く， $n \times n$ の囲碁盤に配置する**連数最大化問題** (Maximum String Problem;MSP) がある．連数最大化問題は矢野らにより，整数計画問題として解かれた．整数計画問題はその問題の性質上，解くのに時間を必要とするので，実時間で使用するには不向きである．ところで，我々は「連数最大化問題」を用いた新しいゲームを提案する予定である．そのゲームにおいて連数最大化問題の最適解を実時間で求める方法を検討した．また，それを求める上で探索木の枝狩りが可能ないくつかの定理を導いた．

Consideration of the Maximum Number of the String in Go

Yuichi Hiruta,[†1] Yohei Yano,[†2] Chikara Igarashi ,[†3]
Yasuhiro Tajima[†4] and Yoshiyuki Kotani [†4]

In recently, there is "Maximum String Problem"(MSP) that puts in a Go board $n \times n$. The problem is putting in a Go board where is arranged more the string of the stone. The problem is solved using Integer Programming Problems(IPP) by Yano. It takes a long time to slove the problem with a property of IPP. We will propose the new game about MSP. Therefore, we have studied how to solve MSP in real time.

1. はじめに

昨年，蛭田により問題提起，電気通信大学の矢野 洋平らにより整数計画問題として定式化された連数最大化問題がある．**第 2 節**で詳しく述べるが連数最大化問題とは，囲碁の升目と碁石（白石，黒石）を用いた問題である．2 種類の碁石をあるルールに従って，囲碁の升目の上に配置する．囲碁の升目で利用する通常の碁盤は 19×19（19 路盤と呼ぶ）で構成されるが，連数最大化問題では一般に $n \times n$（n 路盤）を考える．

連数最大化問題は矢野氏により，0, 1 整数計画問題として定式化されており，19 路盤の通常の碁盤のサイズまで 2006 年 7 月末に解かれている[2)]．

我々は連数最大化問題を用いた新しいゲームを提案する予定である．そのゲームにおいて，強い**ノンプレイヤキャラクタ** (Non Player Character; NPC) を構成するために実時間で連数最大化問題の最適解を解くアルゴリズムが必要である．しかし，整数計画問題では問題の性質上，解くために時間を必要とし，実時間では求めることができない．本稿では，連数最大化問題を整数計画問題ではない別の手法で，高速に実時間で最適解を解く方法を検討する．

2. 連数最大化問題

連数最大化問題 (Maximum String Problem;MSP) を以下に定義する．

2.1 囲碁盤の定義

一般的に通常の囲碁盤は 19×19 の大きさの盤面である．これは 19 路盤と呼ばれる．囲碁には白石と黒石の 2 種類の碁石がある．囲碁はオセロとは異なり，碁石を囲碁盤における線の交点に置きゲームを進める．一般に $n \times n$ の n 路盤というものを定義する．

本稿では n 路盤における左上の座標値を原点とし $(x,\ y) = (0,\ 0)$ と定義する．水平方向は x，垂直方向は y で座標を取ると n 路盤には，座標値が $(0,\ 0)$

†1 東京農工大学大学院 工学府 情報工学専攻
Dept. of Computer and Information Sciences, Graduate School of Technology, Tokyo University of Agriculture and Technology
†2 電気通信大学大学院 電気通信研究科 情報工学専攻
Dept. of Computer Science, Graduate School of Electro-Communications, University of Electro-Communications
†3 東京農工大学大学院 工学府 電子情報工学専攻
Dept. of Electronic and Information Engineering, Graduate School of Engineering, Tokyo University of Agriculture and Technology
†4 東京農工大学 共生科学技術研究院 システム情報科学部門
Dept. of Computer Science, Tokyo University of Agriculture and Technology

から $(n-1,\ n-1)$ まで全 n^2 個の交点が存在する．議論の際にその座標値を利用する．

2.2 連の定義

囲碁における**連** (string) の定義は「おなじ色の石が縦横につながっている」ものである．また，一つの連を構成する石の連なる個数を**連のサイズ** (string size) と呼ぶ[1]．連のサイズは連を構成する石の数というところに注意したい．連と連のサイズの一例を**図 1** に示した．

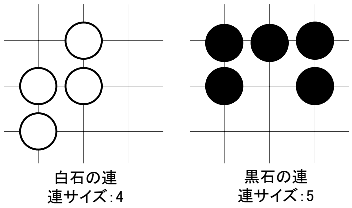

図 1 連と連のサイズの一例

2.3 囲碁のルール

囲碁のルールは：

(1) 2 人で交互に黒と白の石を碁盤の交点に打つ
(2) 呼吸点の無い石は盤上に存在できない
(3) 呼吸点のなくなるところへは打てない
(4) 同型反復の禁止
(5) 勝敗は地の多少で決める

であるが，(1), (3), (4), (5) はこの問題においては直接は関係なく，(2) のみがこの問題において重要である．

呼吸点については，2.4 **節**で詳しく述べる．

2.4 呼吸点

呼吸点 (breathe point) とは，$(i,\ j)$ の石が置いてある交点 $c_{(i,j)}$ にとって上下左右の場所：

- $(i+1,\ j)$ の交点 $c_{(i+1,j)}$
- $(i-1,\ j)$ の交点 $c_{(i-1,j)}$
- $(i,\ j+1)$ の交点 $c_{(i,j+1)}$
- $(i,\ j-1)$ の交点 $c_{(i,j-1)}$

のいずれかに，石の置いていない交点がある場合，その交点を呼吸点と呼ぶ．盤上の角や辺の盤外は，呼吸点にはなれない．囲碁では呼吸点の無い石は盤面に存在することができない．

2.5 連数最大化問題

連数最大化問題 (Maximum String Problem;MSP) とは，n 路盤に存在できる連の数（連数）の最大値を求める問題と定義する．ただし，囲碁のルールに反する盤面は認められない．しかし，囲碁のルール全てではなく：

- 白石，黒石は同じ数でなくても良い
- 呼吸点の無い石は盤上に存在できない

の 2 つが中心的なルールとなる[3]．

連数最大化問題の 4 路盤の解の一例を**図 2** に示した．

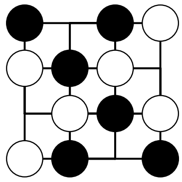

図 2 連数最大化問題の解の一例

3. 問題解決の準備

3.1 市松模様の盤面

3.1.1 市松模様のパターン

碁盤上における市松模様を定義する．碁盤内の任意の交点 (i,j) について，以下の 2 通りが存在する．

(1) 白色基底：

$$\begin{cases}(i+j) \mod 2 = 0 \Longrightarrow \text{白色}\\(i+j) \mod 2 = 1 \Longrightarrow \text{黒色}\end{cases}$$

(2) 黒色基底：

$$\begin{cases}(i+j) \mod 2 = 0 \Longrightarrow \text{黒色}\\(i+j) \mod 2 = 1 \Longrightarrow \text{白色}\end{cases}$$

白色基底，黒色基底のイメージを**図 3** に示した．

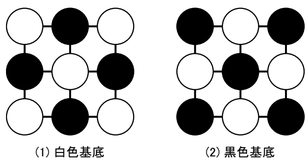

図 3 市松模様のパターン

3.1.2 市松模様化定理

次の定理が矢野により証明されている[2]．

> **定理 1 (市松模様化定理)**．連数最大化問題における最適解の中には，市松模様 *(白黒が交互に並んでいる形)* の盤面が必ず存在する．

証明．任意の盤面は次の作業をすることによって，連数を変化させることなく市松模様状の盤面に変形できる．

(1) 連のサイズが 2 以上の連のサイズを 1 にする
(2) 市松模様と合っていない色を反転する

最適解である盤面も，この方法で市松模様に変形できるので，市松模様の最適解が必ず存在する． □

従って，市松模様に限定して，答えを探すことにより，より効率的に連数最大化問題における解が求まる．ただし，市松模様に限定しているので，連数の最大数は求まるが，その組み合わせが何通り存在するかは求めることができない．

3.2 自由空間と制限空間

n 路盤 ($n \geq 5,\ n \in \mathbf{Z}$) の辺や角により呼吸点が制限されない任意のサイズの盤面の部分集合を**自由空間** (free space) と呼ぶ．一方，辺や角により呼吸点が制限される任意のサイズの盤面の空間を**制限空間** (limited space) と呼ぶ．ここで合法的な，自由空間や制限空間のパターンの集合を考える．自由空間の集合 $\mathbf{B_f}$，辺の制限空間の集合 $\mathbf{B_e}$，角の制限空間の集合 $\mathbf{B_c}$ と定義する．

新たに空間の尺度として，辺により呼吸点の制約があるか，上下左右のいずれに制約がない自由が与えられているかを示す値として，**自由度** (free degree) を定義した．自由空間は 4 方面の自由が与えられているものと考えられ，自由度は 4 と表せる．辺の制限空間は，制限がある 1 辺を含むので，3 方面の自由が与えられているものと考えられ，自由度は 3 と表せる．また，角の制限空間は，制限がある 2 辺を含むので，2 方面の自由が与えられているものと考えられ，自由度は 2 と表せる．上述した関係よりも分かる通り，自由空間の集合 $\mathbf{B_f}$，辺の制限空間の集合 $\mathbf{B_e}$，角の制限空間の集合 $\mathbf{B_c}$ の関係は，

$$\mathbf{B_c} \subseteq \mathbf{B_e} \subseteq \mathbf{B_f}$$

である．また，集合と濃度（大きさ）の関係から，

$$|\mathbf{B_c}| \leq |\mathbf{B_e}| \leq |\mathbf{B_f}|$$

となり，自由空間の集合が一番多いことが分かる．

自由空間や制限空間は格別に大きさの定義の意味合いを含まない．従って，n 路盤における部分集合となりえる $m \times m$（$n > m$）で定義可能であるが，本稿では便宜上 3×3 の自由空間や制限空間を用いる．

自由空間と制限空間の一例を**図 4** に示した．

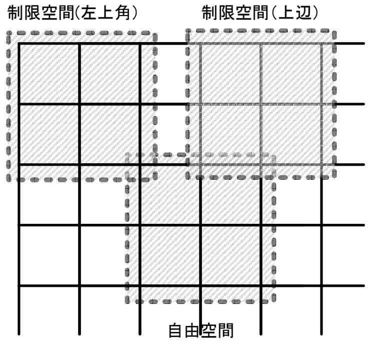

図 4 自由空間と制限空間の一例

4. アルゴリズム設計

4.1 市松模様化定理による効率化

任意の n 路盤で連数最大化問題における解の石の配置は，その任意の 3×3 の空間を取ってきたとき，3×3 の自由空間の全ての合法的な組み合わせのいずれかになっていることは自明である．ここでいう合法的とは，囲碁のルールに則っているということである．

また，囲碁盤の辺や角の制限空間は自由空間に比べて，呼吸点の制約があり自由度が下がる．そのため，制限空間より自由空間の方が，数が多くなる．

本稿では，自由空間，制限空間は，3×3 の空間とした．従って，本来ならば：

- 石を置かない状態
- 黒石を置く状態
- 白石を置く状態

の 3 状態を考えなければならない．3×3 の空間で考えると，故に最大で $3^9 = 19683$ のパターンを調べ，呼吸点のルールに反するものを振るい落とせば良い．

しかし，**定理** 1 により市松模様に限定できる．予め市松模様のパターンを決めておけば，石を置くか置かないかに帰着できる．つまり：

- 石を置かない状態
- 石を置く状態

の 2 状態を考えれば良い．故に最大で高々 $2^9 = 512$ のパターンを調べ，それらの組み合わせから，呼吸点のルールに反するものを振るい落とせば良い．

4.2 パターンマッチングによるアルゴリズム

3×3 の空間の自由空間と制限空間の種類を，**図 5** に列挙した．

- 制限空間（角）
  - 左上角，左下角，右上角，右下角
- 制限空間（辺）
  - 上辺，左辺，右辺，下辺
- 自由空間

の 9 種類である．

実際にシステムにより求めた結果では，自由空間の集合の濃度は 249 通りの配置パターンがあり，制限空間（辺）の集合の濃度は 225 通りの配置パターンがあり，制限空間（角）の集合の濃度は 179 通りの配置パターンがあることが分かった．

これらの制限空間・自由空間を順番に囲碁の盤面の端から置きながら，**走査** (scaning) していくことにより，条件を満たしておける組み合わせを探す．走査の際には，3×3 の空間の左上の座標を (0, 0) としたときの，真ん中である (1, 1) を基準である**走査点** (scanning point) として考える．すなわち，初期ノードの決定時には，盤面の (1, 1) の座標を決め，盤面を (2, 1), (3, 1), $\ldots$, ($n-2$, 1), (1, 2), (2, 2), $\ldots$, ($n-2$, $n-2$) のように順々に走査することにより各位置（走査座標値と呼ぶ）における相応しい配置が，それぞれの走査座標値に対応する空間の集合の中にあるかを調べていく．走査に行き詰まったら，バックトラックで成功しているところまで戻る．走査座標値とそれに対応する空間の

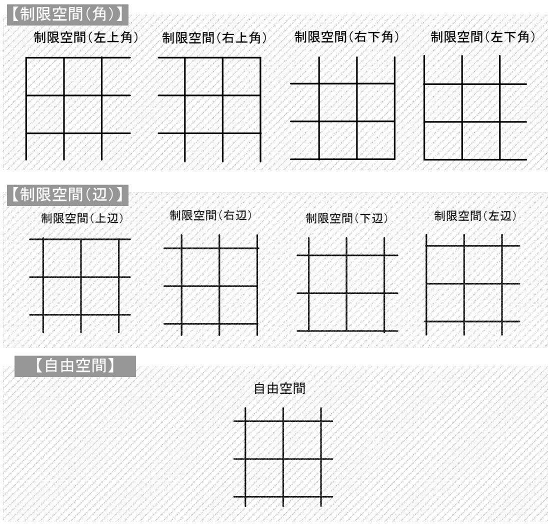

図 5 自由空間と制限空間の種類

集合を**表 1** に示した．囲碁のルール上問題の無い完成できた配置の仕方の中から，連数が最大になっているものを選ぶという深さ優先探索を利用している．

表 1 走査座標値に対応する空間の集合

走査座標値	参照する空間の集合
$(1,1)$	制限空間（左上角）
$(2,1),..,(n-3,1)$	制限空間（上辺）
$(n-2,1)$	制限空間（右上角）
$(1,2),...,(1,n-3)$	制限空間（左辺）
$(2,2),...,(n-3,n-3)$	自由空間
$(2,n-2),..,(n-2,n-2)$	制限空間（右辺）
$(n-2,1)$	制限空間（左下角）
$(n-2,2),...,(n-2,n-2)$	制限空間（下辺）
$(n-2,n-2)$	制限空間（右下角）

走査のイメージを**図 6** に示した．

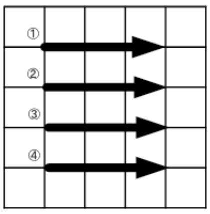

図 6 走査

走査までのパターンマッチングのイメージを**図 7** に示した．

4.3 連数の上限値による枝刈り

4.3.1 石配置上限定理

2.5 **節**で述べているとおり「**呼吸点の無い石は盤上に存在できない**」というルールがある．そのルールの対偶により，以下の定理が導かれる．

> **定理 2 (石配置上限定理).** n 路盤の盤面には，最も効率よく置けたとしても $\dfrac{4}{5}n^2$ 個の石しか合法的に置けない．

証明. 既に述べたとおり,「**呼吸点の無い石は盤上に存在できない**」という性質の対偶により,「全ての盤上の石は，いずれかの呼吸点に属している」ことになる．すなわち，ひとつの呼吸点を中心に考えると，その呼吸点により存在できる石は 最大 4 のみである．つまり,「呼吸点により存在できる石は 最大 4」を図で表してみると，**図 8** である．ここでは，図 8 は，正しい囲碁の置き方では無いが，直感的に理解できるように交点をマスに見立て示した．

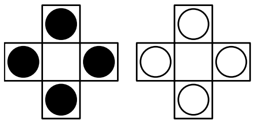

図 8 呼吸点を中心に考えた石の配置

石を n 路盤において最大に配置できる数を C_n とし，そのときの呼吸点の数を b_n とすると：

$$\max C_n = n^2 - \min b_n$$

と書き表すことができる．

すなわち，n 路盤に置いて最大に配置できる数 C_n を求めることは，呼吸点の数を最小することに表裏一体であることが分かる．呼吸点の数を最小にするということは，呼吸点が無駄なく存在するようにすれば良いので，5 個の交点を利用して，4 個の交点に石を置く置き方が一番望ましいことが分かる．従って，n 路盤における n^2 の交点での上限値 $\sup C_n$ ☆は：

$$\sup C_n = \frac{4}{5}n^2$$

であることが分かる．故に，多く石がおけたとしても，高々 $\dfrac{4}{5}n^2$ 個である． □

4.3.2 連数上限定理

連を構成する石の数を連のサイズと呼ぶが，合法的な最適盤面における連のサイズがどのようになるべきかを考える．**定理** 1 より次の補題が導ける．

> **補題 3 (連サイズ最小定理).** 合法的な最適盤面には，連のサイズ 1 のみで構成される盤面が存在する．

証明. **定理** 1 の市松模様化定理より任意の盤面は，連数を変化させることなく市松模様状の盤面に変形できる．仮に連のサイズが 2 を含む，任意の最適盤面が存

☆ sup は数学記号で，上限の意味 (supremum)．$\sup S$ で S の上限．

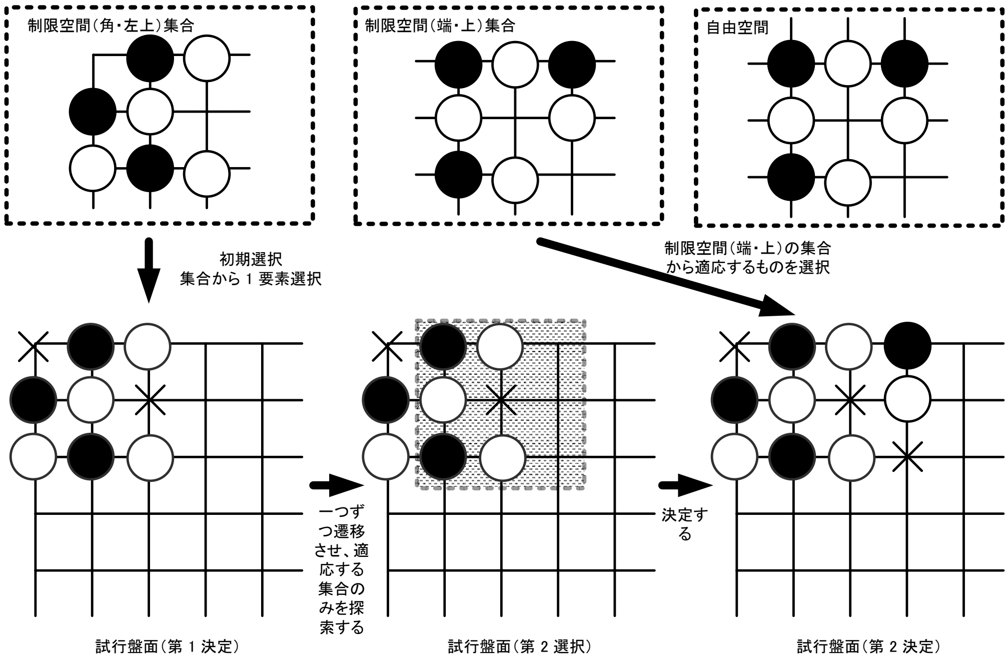

図 7 パターンマッチング

在した場合，その盤面は**定理** 1 により連のサイズが 1 の盤面に連数を変えずに変形可能である．従って，合法的な最適盤面には，連のサイズ 1 のみで構成される盤面が存在する． □

また**定理** 2 の石配置上限定理を発展させると，以下のような定理が導ける．

> **定理 4 (連数上限定理)**. n 路盤の盤面には，最も効率よく置けたとしても $\frac{4}{5}n^2$ 個の連数しか合法的に存在できない．

証明. **補題** 3 の連サイズ最小定理により，最適盤面の中には連のサイズ 1 のみで構成される盤面が存在する．また，**定理** 2 の盤上石配置上限定理により，n 路盤の盤面には，最も効率よく置けたとしても，$\frac{4}{5}n^2$ 個の石しか存在できない．すなわち：

$$\text{石の数} = \text{連サイズ 1 の連数}$$

であるので，n 路盤の盤面には，最も効率よく置けたとしても $\frac{4}{5}n^2$ 個の連数しか合法的に存在できない． □

4.3.3 連数の上限値による枝刈り

パターンマッチングにより初期ノードから順々に配置パターンを決めることを 4.2 **節**で述べた．そこで，連数の上限値を用いた枝刈りが可能である．n 路盤の交点の数 n^2 個のうち，t 個の交点の試行が終わっており，t 個の中で現在 s 個の連数であるとする．そこで，既に敷き詰め終わっている仮の連数値 $s'_{\max}$ が存在するとき，**定理** 4 より残りのマス $(n^2 - t)$ に最も効率よく連を置けたとしても高々$\frac{4}{5}(n^2 - t)$ 個だけであるので，現在までに求まっている連数 s 個と合わせても仮の連数値 $s'_{\max}$ を超えることができない場合，すなわち：

$$s'_{\max} > s + \frac{4}{5}(n^2 - t)$$

であれば，その途中までの配置パターンは失敗である．何故ならば，t 個の交点の試行後，残りをどんなに効率よく敷き詰められたとしても，既にもとまっている仮の連数値に及ぶことができないからである．このように本システムでは連数の上限値という知識を用いた枝刈りを行った．

4.4 深さ優先探索による計算効率化の可能性

通常のゲームやパズルの探索木とは異なり，4.2 **節**のような探索を行えば，連数最大化問題はその問題の特性上，計算過程の依存性が低いので，探索において効率化が図れると考えた．すなわち，**図 9** に示したような左上半分が走査済みで，右下半分が深さ優先により左上半分が違う組み合わせのときに全部の解が計算できているとすると，次の走査地点で残りの連数の最適解が事前に計算した計算より導かれるので，無駄な計算をしなくても効率的にできるというものである．

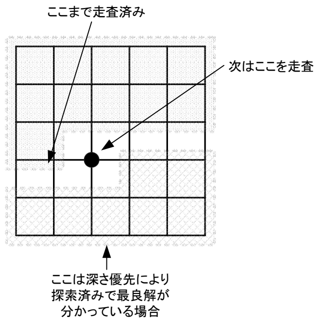

図 9　深さ優先探索による計算効率化の可能性（間違い）

5. 考　　察

5.1　単純な深さ優先探索での組み合わせ爆発

走査座標値の走査の手法で分かるように，走査が探索木の深さになっている．3×3 の空間で走査した場合，非合法まで含めて考えた場合，3×3 の空間の置き方 $p = 2^9 = 512$ を $q = (n-2)^2$ 回走査することになるので，上限は：

$$\sup p^q = 512^{(n-2)^2}$$

となる．単純に走査していたら，たかだか 5 路盤であっても：

$$\sup_{n=5} p^q = 512^{(5-2)^2} = 2417851639229258349412352$$

となり，組み合わせ爆発を起こす．従って，如何に 3×3 の空間の置き方 p を減らし，不要な盤面を探索しないかが焦点となる．

5.2　深さ優先探索による計算効率化の可能性の検証

4.4 **節**に深さ優先探索による計算効率化の可能性を示唆したが，実際は考えに誤りがあった．**図 10** に示したとおり，石の配置は既にパターンマッチングを終えた領域のみ決定しているのではなく，走査の過程で 3×3 の空間を利用しているので，図における $(2, 4)$ を探索しているときには既にその周辺の 1 区画は決定していることになる．すなわち，図において，■で示した領域は既に決定している領域であり，他の石に置き換えることはできない．すなわち，依存関係なしには，それ以降の価値が決まらない．従って，依存関係と価値をまとめて扱う必要があるが，依存関係の数が爆発して効率化が望めないことが分かった．

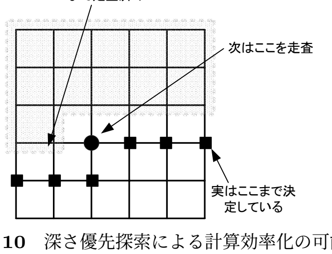

図 10　深さ優先探索による計算効率化の可能性

6. 現状と今後の予定

考察で述べたように，この手法では実時間で最適解を求めることが難しそうだということが分かった．結論としては，今回の研究では，ベストなアプローチに至ることができなかった．従って，今後は別のアプローチで最適解を実時間で求めるようなアルゴリズムを考案したい．

また，これを利用したゲームも，発表する予定である．

参　考　文　献

1) 泉谷 政憲「はじめて覚えるやさしい囲碁入門」池田書店, (1993).
2) 矢野 洋平「連数最大化問題を解くアルゴリズム」電気通信大学 情報工学科 卒業論文（2005）.
3) 「囲碁のルールについて」`http://hp.vector.co.jp/authors/VA003590/rule/igo.htm`