

駒の役割を考慮する詰将棋の生成及び改良手法の研究

佐々木 宣介^{†1} 今田 鈴音^{†1, †2} 長峯 練^{†1}

概要: 将棋 AI は指し将棋において人間を超えるという一つの目的を達成した。現在は強さを探求する将棋 AI だけでなく、様々な用途への活用が模索されている。コンピュータプログラムを用いた詰将棋生成については、市販の将棋プログラムなどでも活用されていると想定されるが、学術研究として発表された文献は多くない。本研究では、コンピュータプログラムによる詰将棋自動生成手法に関して有用と考えられる「駒の役割」という考え方を整理・分類し、それを利用して詰将棋の自動生成を行った結果を報告する。

キーワード: 詰将棋, 創作, 自動生成

A Study of Creation and Improvement Method of Tsume-Shogi Using the Role of Pieces

NOBUSUKE SASAKI^{†1} RION IMADA^{†1, †2} REN NAGAMENE^{†1}

1. はじめに

近年のコンピュータ将棋ソフトウェアの発展は目覚ましく、数年前にはプロ棋士を超えるレベルに到達している。将棋ソフトウェアは、その強さを求める研究だけでなく、様々な用途に用いるプログラムの研究も模索されるようになっていく。

詰将棋におけるコンピュータの活用として、詰将棋解答プログラムは比較的早い段階から成果があがり、1997年には脊尾のプログラムにより、最長手数詰将棋「マイクロコスモス」を初めてコンピュータで解くことに成功し[1]、さらに2002年の長井らのdf-pnアルゴリズム[2]により、ほぼすべての詰将棋を解くことが可能となった。現在、詰将棋の創作においては、余詰検査の機能を備えたソフトウェアを活用して、余詰等のない完全作の検証に活用されている。

一方、詰将棋創作をコンピュータプログラムが行うことについては、学術論文として発表されたものはそれほど多くない。本研究では、コンピュータによる詰将棋創作に寄与する手法として、駒の役割を利用する詰将棋創作手法の提案と検証を行うものである。本手法を用いることにより、詰将棋創作における「手順法」による創作を行うことにつながると期待できる。

2. 関連研究

2.1 詰将棋とは

詰将棋とは、将棋のルールを用いたパズルの一種である。駒が配置された初期局面から、王手の連続で相手の玉を詰

ませることを目的とする。芸術性のあるパズルとして、いわゆる指し将棋から独立した一つの分野となっており、多くの将棋ファンからも愛好されている。

以下に詰将棋に関する重要な用語を示す。

詰み: 本来ならば玉方の手番であるが、玉方に王手を外す指し手が存在しない状態。指し将棋でいうと玉方が負けの状態。

不詰: 正解手順がないこと、つまり、解がないこと。

攻方: 玉を詰ませることを目的とする側の手番。先手。できるだけ短手数で詰むような手順を選択する。

玉方: 玉が詰まされるのを防ぐことを目的とする側の手番。後手。詰将棋では最も詰みまでの手数が長くなるような手を選択する。ただし、手数稼ぎになる無駄な合駒（無駄合という）はしないこととする。

余詰: 攻方に正解手順以外の玉を詰ませる手順があること。つまり、解が2通り以上あること。ただし、例外として最終手（最後の1手）は考慮しない。

駒余り: 詰み局面で攻方に持ち駒が残ること。

完全作: 解答手順が存在し、余詰、駒余りがないこと。

プレイヤは攻方、玉方それぞれに課せられたルールに則った最善の指し手を想定しながら解答手順を特定する。

2.2 詰将棋の創作手法

詰将棋の創作には、大きく分けて2つの手法がある。ある局面について、詰将棋として成立するように調整を加えることで問題を創作する「順算法」と、詰み局面から、その局面に至るように手数を伸ばして逆算し生成することで詰み手数を伸ばす「逆算法」と呼ばれる手法である。また、

^{†1} 県立広島大学
Prefectural University of Hiroshima

^{†2} (現) 株式会社ドリームオンライン
Present affiliation is DreamOnline Co., Ltd.

順算法に分類されると考えられるが、作者が意図した手順を中心に考え、その手順が実現するように駒の配置等を調整していく「手順法」と呼ばれる創作手法もある。

これまでのコンピュータにより詰将棋を自動生成する試みには、以下のようなものがある。

まず、広瀬らによる、逆算法を用いた局面の生成を繰り返し行うことで問題を生成する手法[3]がある。詰み局面から開始し、生成した局面のうち完全作であるものを逆算の対象とすることで、完全作であることを維持しながら手数伸ばしている。また、野下は簡約関係という概念を取り入れた駒の削除と変更による問題生成を行った[4]。また、加藤により詰将棋に関する各種情報発信がされており[5]、詰将棋創作や余詰検査機能の使い方など充実した情報が提供されている。最近では、宗藤らによる遺伝的アルゴリズムを用いて長手数の問題を生成する手法[6]などがある。

また、詰将棋作品の芸術性に着目した評価をコンピュータが行う手法については、小山らによる詰将棋の感性評価に関わる要素を定量的に分析する試み[7]や、石飛らによる問題を解く際の証明数・反証数を完成評価に利用する試み[8]などがある。

本研究では、特定の手順を維持しながら駒の置き換え等の盤面更新を行うことをある程度意識しながら行う手法の提案とシステムの実装を行うものである。この手法を用いることで「手順法」のような詰め手順を維持しながら局面の変更を行っていく創作につながる可能性がある。

3. 詰将棋における駒の役割

ある局面で駒配置を変更した場合、その局面の詰み状態と、それ以降の指し手の成立に影響を与える。例えば、詰み局面で王手している駒の位置を変更すれば、その局面は詰み局面ではなくなる。そのような詰将棋における駒が果たす役割について、分類を行った。

まず、局面の詰み状態に影響する役割として以下の4種類を設定した。これらの役割は、盤上のマス目ごとに評価される。

- 王手
- 玉の逃げ道をふさいでいる
- 王手駒を取ることができる
- 王手を防いでいる

さらに、詰み手順における指し手の成立に影響する役割として、以下の4種類を設定した。

- 移動可能である
- 移動を防いでいる
- 取ることができる
- 取られることができる

「移動可能である」と「移動を防いでいる」ことの例を図1の左側に示す。攻方の6八角は、王手として4六に移動することが可能である。この場合、6八角は▲4六

角という指し手に対して移動可能であるという役割を果たす。一方、もし7七に移動することができれば王手となるが、実際には7七に移動することはできない。この時、7七の歩が、「移動を防いでいる」という役割を果たしている。

図1の右は、「取ることができる」役割と「取られることができる」役割の例である。この図は、▲2一金△同玉▲2二金という3手詰の局面であるが、2一の金は初手からの▲2二金という1手詰を防ぐ玉方の駒としての役割を持っている。しかし、3手目では王手の役割を果たさなければならない。つまり、1手目では玉方としての役割、3手目では攻方としての役割を果たすことになる。このような役割を実現するのが「取られることができる」役割である。

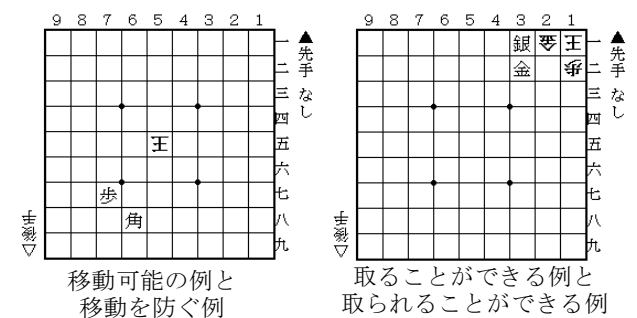


図1 指し手の成立に影響する役割の例

Figure 1 The Example of Roles Affecting of the Moves.

4. 開発システムの概要

4.1 システムの概要

本研究で作成した詰将棋生成・改良システムについてその概要を述べる。本システムは、余詰のある詰将棋を初期局面の入力として与えると、駒の役割を考慮した駒の置き換え、無作為な駒の追加等の処理を通じて、さらに評価の高い手順が見つかるように局面を更新する処理を繰り返し、手順がより長い、余詰が少ないなど、より評価の高い詰将棋の作成を行うものである。

4.2 駒の置き換え

入力局面を改良するための方策として、解答手順・余詰手順の中で最も優れた手順を「優先手順」として設定し、なるべく優先手順を維持しながら他の解答手順・余詰手順を消去することを考える。

駒の置き換えの際は、前述の駒の役割の考えを用いて、現在の正解手順に関係する駒の役割を全ての手順から抽出する。続いて、それらの役割を維持しながら、盤上の駒が他の駒に置き換えることが可能であるか確認する。置き換えることが可能であれば置き換えにより、その局面がこれまでより評価の高い局面になるかどうか確認し、評価の高いものであれば、置き換えを行う。

図2に駒の置き換えの例を示す。玉の逃げ道をふさいでいる役割を持つ香を金に置き換えることができている。つまり、詰み手順を変えずに駒の置き換えができることになる。

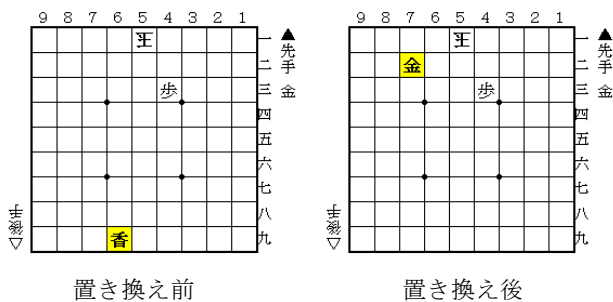


図 2 駒の置き換えの例

Figure 2 The Example of Replacement of Piece.

4.3 駒の追加

駒の削除と置き換え処理のみでは、盤面から駒が減るだけで増えることはない。そのため、より多くの詰将棋局面を生成するために、駒を追加する処理も行うようにした。

駒の追加は、後手の駒台にある駒を無作為に 1 種類選び、盤上もしくは先手の駒台に追加する。駒の追加には、「(1)攻方の持ち駒とする」、「(2)攻方の盤上の駒とする」、「(3)玉方の盤上の駒とする」の 3 種類のケースが考えられるが、その時点の余詰の数によって、以下のように選択した。複数の追加方法のどれを選択するかはランダムに決定している。

盤上に駒を置く場合、作品局面に一定の影響を与えやすい位置のみを配置候補位置とするようにした。各駒種それぞれに以下のように設定している。

- 飛車・龍**：玉のいる筋と段の上下左右 2 列・2 段の範囲
- 角・馬**：玉のいる位置から斜めに見て前後 2 列の範囲
- 香**：玉の筋とその左右 2 列。なるべく多くの利きを持つように下段から置くようにする
- 金・銀・歩・と金**：玉の位置から 2 マス以内の 24 マスの範囲
- 桂**：攻方の場合は前述の金等の範囲の玉の手前の 8 マス、玉方の場合は、玉の奥の 8 マス

図 3 に角と香車の例を示す。

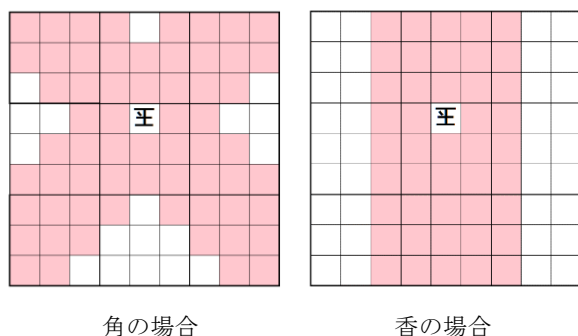


図 3 駒の追加候補位置の例 (角と香の場合)

Figure 3 The Example of a Piece Addition Candidate Position (The Example of “Kakugyou” and “Kyousha”).

表 1 駒の追加方法

Table 1 The method of Adding Piece.

余詰の数	駒の追加方法
0	(1), (2), (3)
1	(1), (2)
2~3	(2)
4 以上	(3)

4.4 システムの処理の流れ

開発したシステムでは、余詰のある詰将棋局面を入力局面とする。入力局面に対して駒の追加・削除・置き換えにより、詰み手数を伸ばす、あるいは余詰手順の数を減らすといった改良を繰り返していく。本システムは、局面が入力されたら、以下の流れで詰将棋局面の改良を行う。

- 1 つもしくは複数ある詰手順から、一番評価の高い手順を「優先手順」と設定する。詰手順評価は「詰み局面で余る駒が少ない」、「手順が長い」という順番で評価する。
- 優先手順中の全手順で駒の役割を求める。
- 優先手順の中で役割を持っていない駒を削除する。
- 駒の追加を行う。
- 局面全体の役割を調べてもう一度駒の削除を行う。
(駒の追加が実際には局面に影響を与えていない場合や、駒の追加により、役割の競合のため役割を失った駒が削除される)
- 役割を考慮した駒の置き換え処理を行う。
- この過程の中で、それまでの詰将棋局面よりも評価の高い局面が得られた場合に、最初に戻り、優先手順の設定からやり直し、局面の更新を繰り返す。

詰将棋局面の評価では、以下の要件を優先順位として設定し、評価が高くなるようにした。

1. 余詰が減る (余詰が 3 より多い場合)
2. 詰み手数が伸びる
3. 余詰が減る (余詰が 3 以下の場合)
4. 後手の駒台を除く駒の総数が増える

新たに生成された局面が、このいずれかの要件が改善されて最高評価作が更新されたら、同時に優先手順の決定も再び行い、改良処理を継続する。手数が伸びることと余詰を減らすことの両方を評価しているが、余詰が多い場合には余詰を減らすことを優先し、余詰が少なくなった場合には手数を伸ばすことを優先する。

優先手順の決定や局面の評価の際は、余詰等の検査を行うが、その検査は詰将棋解答・余詰検査ソフト「柿木将棋 IX」[9]を用いて行った。柿木将棋の余詰検査は、時間制限 60 秒で行った。また、システムの動作条件として、一定以上の手数まで詰手順が延長された場合にシステムを終了するように設定したが、実際にはそこまで至らずに停滞する状況であったため、システムを約 5 時間程度以上動作させた上

で、局面の改良が進まない場合に手動で実験を打ち切り、出力局面とした。

5. 開発システムの評価

プロ棋士の実戦やインターネット上の作品などから得た余詰のある局面または詰将棋を用いて 40 通りの局面を準備した (タイプ A)。また、余詰がある簡易的な詰将棋の局面を 5 通り準備した (タイプ B)。タイプ A は、実戦を題材にしたため、複雑な局面が多い。そのため駒の追加機能による幅広い局面の生成が有効に働きにくいと考え、比較的簡単な局面からスタートするタイプ B の入力についても実験を行った。また、タイプ B については、出力局面のバリエーションがどの程度広がるかを確認するため、同じ初期局面を入力とする実験をそれぞれ 6 回実施した。

それぞれの入力局面およびシステムが出力した局面の状況をまとめたデータを表 2 から表 5 に示す。(表 5 において例えば「B-1-2」という記述は、B-1 を初期局面とした 2 回目の実験結果のことを示している)

どちらの局面群においても、大部分の問題で局面の更新が行われ、その結果として詰手数の延長に成功している。局面の更新が 1 回もできなかったのは、A-11 及び、B-3 の 4 回目の 2 つだけで、それ以外は何らかの更新が行われたことが分かる。また、余詰が 0 となるだけでなく、駒余りもない完全作 (非限定、手順前後、迂回はキズとして許容することとする) は、A-15, A-19, A-29, A-33, B-1-1, B-1-6, B-2-3, B-2-4, B-2-6, B-3-2, B-3-6, B-4-3 であった。

また、表 5 の結果から、タイプ B の局面からは、同じ初期局面からも、幅広い様々な局面が生成可能であることがわかった。

これらの結果の中から、局面の改善が比較的うまくいったいくつかの例を以下に紹介する。

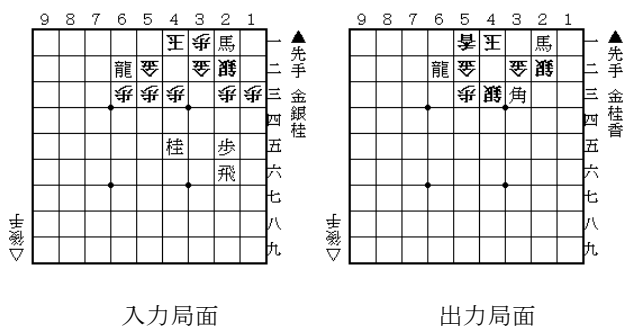


図 4 A-19 の実験結果
Figure 4 The Result of A-19.

図 4 は A-19 の初期局面と出力結果である。初期局面における最短の詰め手順は、▲5 一金△同金▲5 三桂不成までの 3 手詰である。そのため表 2 の詰手数の数値としては 3 手詰と判定されている。しかしシステムの改良処理においては、最短手順・余詰手順を合わせた 5 通りの手順から、▲4 二銀△同金左▲5 三桂不成△同金左▲4 二歩△同玉▲3

四桂△3 三玉▲2 二馬△3 四玉▲2 三馬△同玉▲2 四銀△3 二玉▲3 六飛△2 一玉▲2 二金△同玉▲5 二龍△3 二金▲同飛成△同歩▲2 三金△1 一玉▲1 二金打までの 25 手詰が優先手順として設定された。この手順は駒余りである (歩が 1 枚余る) が、最短手順も駒余りであるため、詰み手数の長さが優先されて優先手順として選ばれた。この優先手順詰み手順は、玉の周りにある金銀三枚の堅い守りを崩していく手順となっている。この入力局面から得られた出力局面では、余詰・駒余りのない完全作となり、解答手順は▲5 一角成△同金▲4 二香△同金寄▲3 三桂△同金▲4 二香△同金▲5 一金までの 9 手詰である。当初の優先手順から手数は減っているが、玉の守りを盤上の大駒を中心に崩していく優先手順の序盤の印象は残った手順となっている。

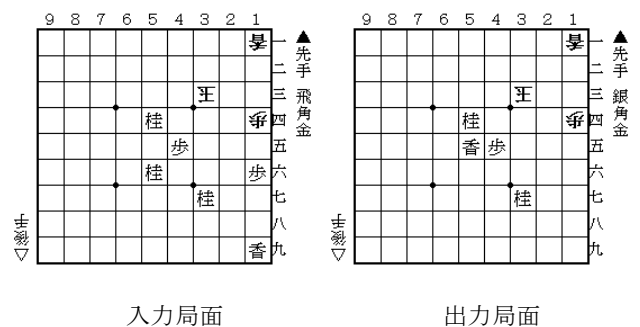


図 5 A-29 の実験結果

Figure 5 The Result of A-29.

図 5 は、A-29 の初期局面と出力結果である。A-29 は、最短の詰み手順 7 手詰に対し、余詰が 33 と非常に多いが、出力局面では 15 手詰の完全作となった。最短手順は 7 手詰であるが、優先手順としては、▲3 五飛△3 四角▲4 四角△2 三玉▲3 三金△1 二玉▲2 二金△1 三玉▲2 五桂△2 四玉▲2 三金△同玉▲3 三角成△1 二玉▲3 四馬△2 三歩▲同馬△同玉▲3 三飛成△1 二玉▲2 三角△2 一玉▲3 二龍までの 23 手詰が設定された。出力局面は、▲5 一角△3 四玉▲2 五銀△2 三玉▲2 四銀△1 二玉▲1 三銀成△同玉▲2 五桂△1 二玉▲1 三金△2 一玉▲3 三桂不成△3 二玉▲4 二角成までの 15 手詰である。この実験においても、駒の追加による詰み手数の延長と優先手順を残しつつ駒を置き換える処理がうまく行われたと考えられる。

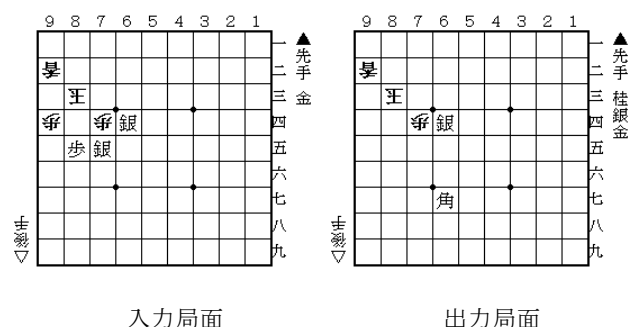


図 6 B-3-2 の実験結果

Figure 6 The Result of B-3-2

図 6 は、B-3 の初期局面と 2 回目の実験の出力結果である。B-3 では 3 手詰のわりに余詰が 4 つと比較的多かったが、この出力局面では 9 手詰と詰み手数が伸びて、余詰も駒余りもない完全作を作成することに成功している。初期局面の最短手順は▲7 三金△9 三玉▲8 四銀までの 3 手詰であり、▲8 四銀△8 二玉▲7 三銀成右△9 一玉▲8 二金までの 5 手詰を優先手順として設定された。出力局面は、▲9 五桂△同香▲7 三金△8 四玉▲8 五銀△9 三玉▲9 四銀△8 二玉▲8 三銀成△8 一玉▲8 二成銀までの 11 手詰で、余詰、駒余りがない作品（ただし、迂回手順になる軽微なキズがある）となっている。

6. おわりに

本研究では、コンピュータによる詰将棋自動生成手法として、駒の役割に着目し、駒の果たしている働きを維持したまま駒の置き換えを行う手法を提案した。この手法を用いれば、手順法のような詰み手順を中心に考えて、その手順の実現を目指す手法にも応用が可能と考えられる。

今後の課題としては、以下の点があげられる。現在のシステムでは、複数の駒を同時に置き換えることができないため、複数の駒による置き換えを行う機能の実装が必要と考えられる。また、今回行った役割の分類をさらに詳細に検討して、効率よく駒の置き換えができるようにすることなども必要と考えられる。さらに、「手順法」という創作手法を意識するのであれば、優先手順の扱いについて、なるべく優先手順を残すことを目指す機能、最初から優先手順を手動で設定する機能などが今後の課題としてあげられる。

参考文献

- [1] 春尾昌宏, “パソコンで詰将棋”, 情報処理学会誌「情報処理」, vol. 39, no. 5, pp. 450-453, 1998.
- [2] 長井歩, 今井浩, “df-pn アルゴリズムの詰将棋を解くプログラムへの応用”, 情報処理学会論文誌, vol.43, No.6, pp.1769-1777, 2002.
- [3] 広瀬正幸, 伊藤琢巳, 松原仁, “逆算法による詰将棋の自動創作”, 人工知能学会誌, vol. 13, no. 3, pp. 452-460, 1998.
- [4] Noshita, K, “A Note on Algorithmic Generation of Tsume-Shogi Problems”, In Game Programming Workshop in Japan '96, pp. 27-33, 1996.
- [5] 加藤徹, “詰将棋メモ”, <http://toybox.tea-nifty.com/memo/>, (参照 2020-02-15).
- [6] 宗藤大貴, 長尾智晴, “進化計算法を用いた詰将棋の自動生成”, ゲームプログラミングワークショップ 2019 論文集, pp. 1-6, 2019.
- [7] 小山謙二, 河野泰人, “名作詰将棋における感性の定量的評価”, 情報処理学会論文誌, vol. 35, no. 11, pp. 2338-2346, 1994.
- [8] 石飛太一, 飯田弘之, “詰将棋問題の完成評価と証明数に関する考察”, ゲームプログラミングワークショップ 2012 論文集, pp. 163-166, 2012.
- [9] “柿木将棋”, <http://kakinoki.o.oo7.jp/kshogi.htm>, (参照 2020-02-15).

表 2 タイプ A の入力局面

Table 2 Initial Position of Type A problems.

局面	初期状態の状況				
	詰手数	余詰	非限定	手順前後	迂回
A-1	7	7	0	0	0
A-2	11	3	0	0	0
A-3	11	23	0	0	0
A-4	9	1	1	0	0
A-5	13	5	0	0	0
A-6	11	4	0	0	0
A-7	11	3	0	0	0
A-8	13	11	0	0	0
A-9	27	1	0	1	0
A-10	13	3	0	0	0
A-11	33	1	1	0	0
A-12	11	4	0	0	0
A-13	15	0	3	0	0
A-14	11	10	0	0	0
A-15	15	0	3	0	0
A-16	13	1	0	0	0
A-17	9	1	0	0	0
A-18	23	9	5	0	0
A-19	3	4	0	0	0
A-20	17	16	1	0	0
A-21	11	16	0	0	0
A-22	5	6	0	0	0
A-23	9	10	5	0	0
A-24	3	2	1	0	1
A-25	5	2	0	0	0
A-26	17	47	1	0	0
A-27	7	7	0	0	0
A-28	7	1	0	0	0
A-29	7	33	2	0	0
A-30	17	6	0	0	0
A-31	11	1	0	0	0
A-32	9	7	0	0	0
A-33	9	0	0	0	0
A-34	21	5	0	0	0
A-35	7	3	0	0	0
A-36	3	6	0	0	0
A-37	9	1	1	0	0
A-38	7	24	0	0	0
A-39	7	0	0	0	0
A-40	11	8	0	0	0

表 3 タイプ A 局面のシステム出力結果

Table 3 Outputs for Type A Problems.

局面	出力状態の状況					
	詰手数	余詰	非限定	手順前後	迂回	改良回数
A-1	35	3	2	0	0	1707
A-2	19	0	0	0	0	32
A-3	15	1	0	0	0	76
A-4	33	3	2	0	0	285
A-5	13	1	0	0	0	196
A-6	13	1	0	0	0	49
A-7	37	1	0	0	0	34
A-8	27	0	1	0	0	49
A-9	27	1	0	1	0	12
A-10	15	1	0	1	0	30
A-11	33	1	1	0	0	0
A-12	17	1	0	0	0	28
A-13	23	0	3	0	0	19
A-14	21	3	0	0	0	154
A-15	17	0	2	0	1	42
A-16	13	1	0	0	0	5
A-17	13	2	1	0	0	360
A-18	23	9	5	0	0	12
A-19	9	0	0	1	0	35
A-20	29	0	3	0	0	18
A-21	21	2	1	1	0	7
A-22	27	2	2	1	1	174
A-23	13	1	0	0	0	30
A-24	9	1	0	0	0	67
A-25	9	0	0	0	0	5
A-26	17	46	1	0	0	6
A-27	9	1	0	0	0	6
A-28	17	3	0	0	0	414
A-29	15	0	1	0	0	342
A-30	25	3	1	1	0	7
A-31	21	3	0	0	0	67
A-32	11	2	0	0	0	31
A-33	9	0	0	0	0	17
A-34	21	5	0	0	0	21
A-35	23	1	0	0	0	46
A-36	5	1	0	0	0	110
A-37	11	3	2	0	0	9
A-38	17	1	1	0	0	1000
A-39	15	3	0	0	0	57
A-40	27	11	5	0	0	5

表 4 タイプ B の入力局面

Table 4 Initial Position of Type B Problems.

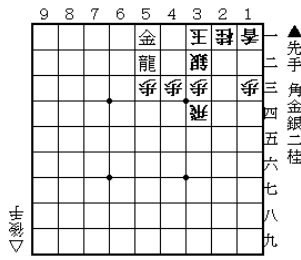
局面	初期状態の状況				
	詰手数	余詰	非限定	手順前後	迂回
B-1	3	1	0	0	0
B-2	3	3	0	0	0
B-3	3	1	0	0	0
B-4	5	5	0	0	0
B-5	3	1	0	0	0

表 5 タイプ B 局面のシステム出力結果

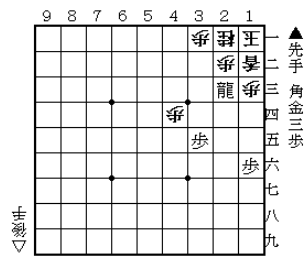
Table 5 Output for Type B Problems.

局面	出力状態の状況					
	詰手数	余詰	非限定	手順前後	迂回	改良回数
B-1-1	5	0	0	0	0	4
B-1-2	11	3	1	0	0	51
B-1-3	7	1	0	0	0	36
B-1-4	7	1	1	0	0	167
B-1-5	9	1	1	0	0	44
B-1-6	5	0	0	0	0	25
B-2-1	11	3	2	0	0	31
B-2-2	27	1	4	0	1	135
B-2-3	9	0	2	0	0	142
B-2-4	7	0	0	0	0	114
B-2-5	11	3	0	0	0	32
B-2-6	13	0	0	1	0	41
B-3-1	13	1	2	0	0	106
B-3-2	11	0	0	0	1	257
B-3-3	19	3	5	0	0	78
B-3-4	5	2	0	0	0	3
B-3-5	13	1	2	0	0	213
B-3-6	5	0	1	0	0	147
B-4-1	9	2	0	0	0	93
B-4-2	7	0	0	0	0	4
B-4-3	9	0	0	0	0	3
B-4-4	13	1	0	0	0	17
B-4-5	7	0	0	0	0	4
B-4-6	5	2	0	0	2	273
B-5-1	7	2	1	0	0	137
B-5-2	17	2	8	0	0	36
B-5-3	11	2	0	0	0	258
B-5-4	3	1	0	0	0	0
B-5-5	9	1	1	0	0	27
B-5-6	21	2	0	0	0	72

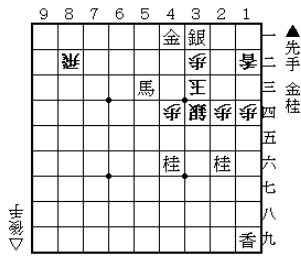
付録 本システムの実験に使用した初期局面



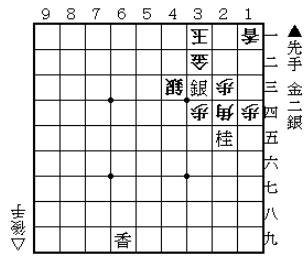
A-1



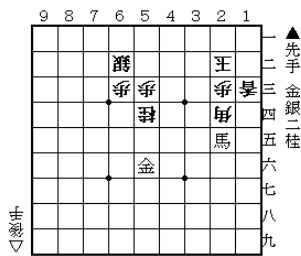
A-2



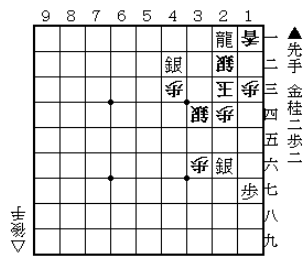
A-3



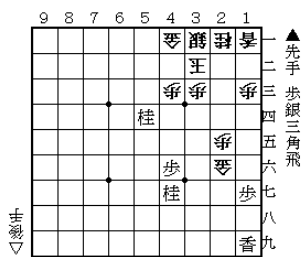
A-4



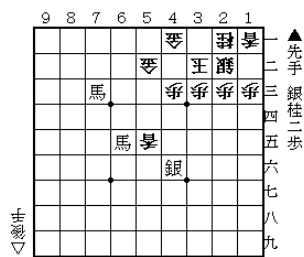
A-5



A-6



A-7



A-8



A-9



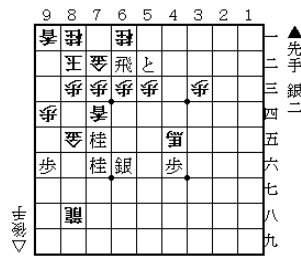
A-10



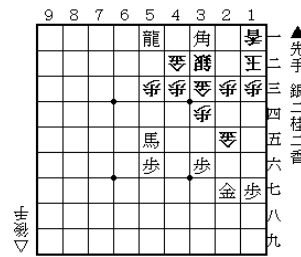
A-11



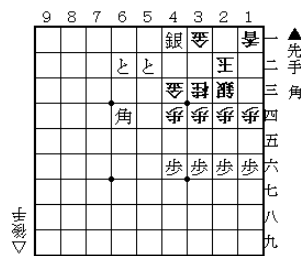
A-12



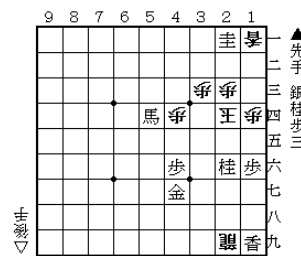
A-13



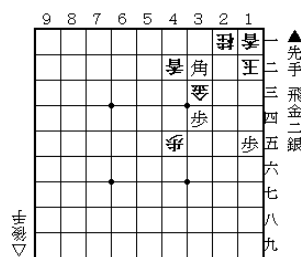
A-14



A-15



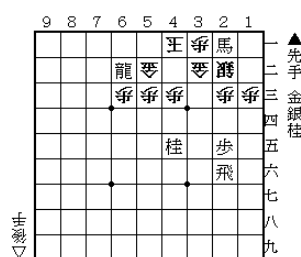
A-16



A-17



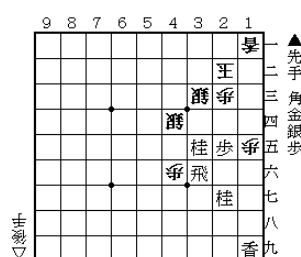
A-18



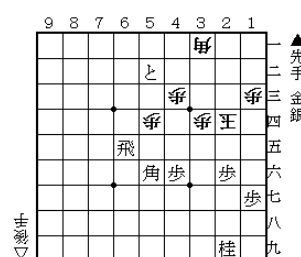
A-19



A-20



A-21



A-22



A-23



A-24

