Worst1.exe 取扱い説明書

2021年9月28日 1.97版用

_	.1. > 10 %	_
	まえがき	
	使用方法	
	(1) コマンドプロンプトからの使用	3
	(2) Kifu for Windows からの起動	3
	(3) プログラムの実行と出力	6
	(4) 手順指定	8
2.	詰将棋情報の設定内容	14
3.	特殊な駒、仕様	17
4.	動作	18
5.	出力ファイル	20
6.	注意(特に普通詰将棋を解く場合)	22
	(1) 普通の詰将棋と「最善詰」の違い	
	(2) 千日手の扱い	23
	(3) 左右対称の扱い	23
7.	トレース、モニタ機能と本プログラムの動作原理	25
	(1) トレース、モニタ機能について	25
	(2) モニタの表示・非表示の切り替えと深度の変更	28
8.	中断・再開・途中経過の表示	29
	(1) 中断	
	(2) 再開	30
	(3) 途中経過の表示	31
	動作に関するオプションについての補足	
	(1) 解図のみ	
	(2) 略式検討	32
	(3) 余詰検討	
). 性能の限界	
	. 連絡先	
). 更新履歴	

0. まえがき

Worst1.exe は最悪詰を検討するために作成したプログラムです。 現在は対象を他の対抗系(協力系でない)ルールにも広げ、少しずつ機能を追加しています。

本ソフトウェアはフリーソフトとして配布します。 無保証であることをご承知の上でお使いください。 再配布も自由に行って構いませんが、必ず本説明書を添付してください。

【注意】本プログラムは検討開始時に無条件に約30MBのメモリを消費します。 また、使用時間の増加と共に追加のメモリも必要とします。 使用時にはメモリ残量にご注意ください。

a) 動作環境

本プログラムは Windows XP 以降で使われることを想定しています。 それ以前 (Windows 2000、98 や Me など) では動作しません。 64bit 版と 32bit 版があり、64bit 版は 32bit 版より多くのメモリを使うことができます。

b) 仕様上の制限

本プログラムでは1手から2999手が仕様上取り扱える範囲です。 また、「連続王手の千日手の禁止」というルールは採用していません。 従って、それを利用した作は解けません。

c) 特殊な機能

本プログラムは検討や創作に資するため、以下の機能を提供しています。

- 「受方持駒制限」をサポートしています。部分的には駒数の拡張もできます。
- ・「動かない、取れない、通過できない」マスを定義できます。
- ・「受先形式」(受方から開始する作)も検討できます。

1. 使用方法

(1) コマンドプロンプトからの使用

本プログラムを適当なフォルダに置き、入力ファイルを kif 形式で作成し、コマンドプロンプトから以下の形式で入力ファイルを指定します。

C:\\ 本プログラムへのパス 入力ファイルへのパス

例えば入力ファイル名を test.kif とし、本プログラムと共に C:\{\text{Ywork}} というフォルダに置いた場合は、コマンドプロンプトを起動してから以下のように操作します。

Microsoft Windows [Version 6.3.9600]

(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Line\cd \Line\work

C:\footnote{\text{work}\range}\text{worst1 test.kif}

コマンドプロンプトを使うのが面倒な場合は、デスクトップ等にショートカットを作成し、入力ファイルをWorst1.exe またはそのショートカットにドラッグ&ドロップしても良いでしょう。

(2) Kifu for Windows からの起動

もっと便利なのは柿木義一氏の作成した Kifuw (<u>柿木の将棋ソフトウェア</u>から入手してください) から本プログラムを呼び出せるようにすることです。



Kifuw のメニューから、[ツール] \rightarrow [ツールメニュー設定]を選び、空いた場所を適当に選んで Worst1.exe を使うよう設定します。

以下は「メニュー2」に登録した例です。

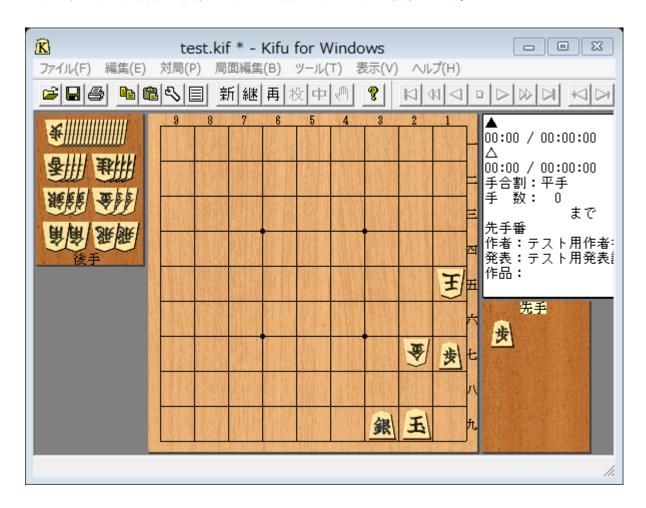
	ツールメニュー設定	EX.			
	J=2 ▼	ОК			
火工~表示:	Worst1.exe	キャンセル			
起動プログラム:	C:¥work¥Worst1.exe	参照(R)			
引数:	"%K"				
%K 棋譜ファイルフルパス %D KIFUW.EXE のパス(¥付き) %N 手数					

「引数」は二重引用符(")で囲ってください。

これで間にスペースが入るようなフォルダに棋譜ファイルが置いてあっても解析することができます。 設定が終わったら実際に何か入力して試してみましょう。

ここでは「最悪自玉詰」でのサンプルを用いて説明します。

まずは、通常の詰将棋の編集と同様の手順で盤面・持駒を編集します。



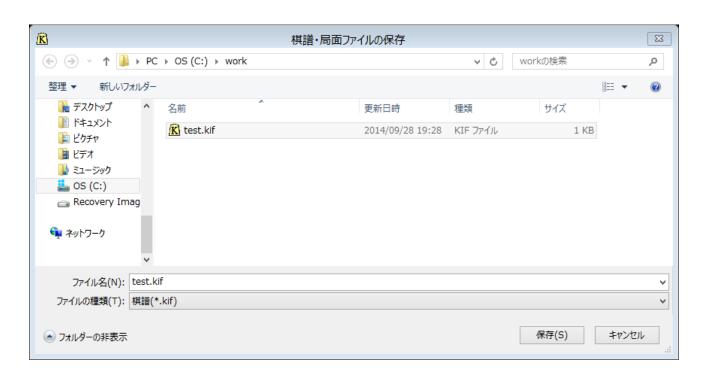
次にメニューの[編集]→[詰将棋情報]を選んで、**手数**と**ルール**を設定します。 ルールは「分類」の項目に入れてください。

	詰将棋情報	X
作品番号:	▼	OK
作品名:	テスト用作品名 🔻	キャンセル
作者:	テスト用作者名 🔻	
発表誌:	テスト用発表誌 ▼	
発表年月:	20XX年XX月 ▼	
出典:	▼	
手数:	6	
完全性:	▼	
分類:	最悪自玉詰	
受賞:	▼	
備考:	<u> </u>	

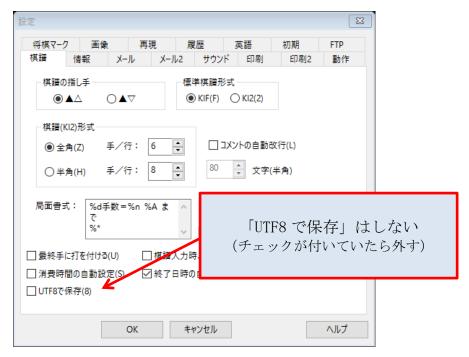
「手数」「分類」以外の項目の入力は任意です。

また、「作品名」「作者」「発表誌」「発表年月」の項目は出力時に1行にまとめて出力されます。 設定が終わったら、適当なファイル名で保存してください。

このときファイル形式は kif か ki2 にしてください。(※bod 形式だと「詰将棋情報」が保存されません)



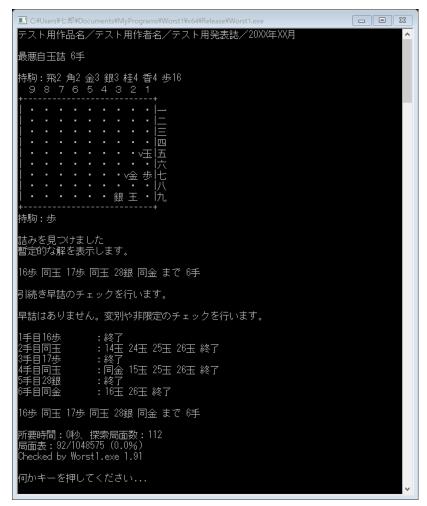
【注意】Kifuw V7 系をお使いの方は、[設定]→[棋譜]の「UTF8 で保存」のチェックを外してください。 Worst1.exe は今のところ UTF8 に対応していません。



保存が終わったらメニューの[ツール]からツールメニューを選んで実行してください。

(3) プログラムの実行と出力

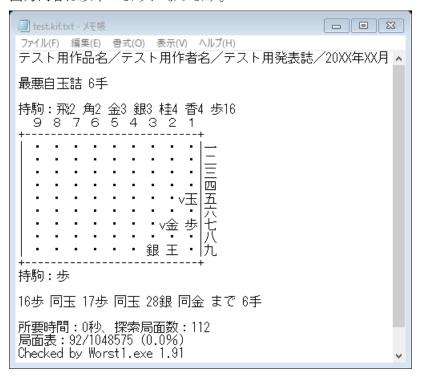
先程の例では「ツールメニュー2」に Worst1.exe を登録したので、これを選びます。



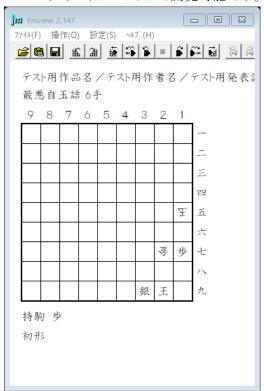
解析が行われ、作意手順が検出されました。

(上の例では見易くするため普通の画面より縦長にしています。)

「何かキーを押してください…」と表示されるので、出力内容を確認したら、Enter キーを押してください。 Enter キーを押すとプログラムは終了し、ウィンドウが閉じます。 結果は元のファイルがあったのと同じ場所に、元の名前に.txt という拡張子を付けて保管されています。 既にファイルがある状態で再度解析をすると追記されます。(※1.50版から追記方式に改めました。) 出力内容は以下のようになります。



このファイルは fmview で閲覧可能です。



入力は Kifuw で閲覧が fmview というのは変則的ですが、特殊駒や非標準の駒数を使う場合に備えてこのような仕様になっています。

【注意】余詰などを検出して配置を修正した場合は、必ずファイルの保存を行ってください。 保存せずに Kifuw 上の配置だけ変えても、検討対象に反映されません。

(4) 手順指定

1.80 版から「手順指定」の検討機能を追加しました。

あらかじめ作意を入力しておくと、その作意を元に検討する機能です。

ただし作意を盲目的に信頼するのではなく、変化が詰むことを確認してから、以降の検討作業を行います。 作意は途中まででも構いません。作意が入力されているところは、変化が詰むことだけを確認し、作意が途 切れたところから通常の解図と検討の作業を行います。

変化が詰まない場合、作意以外の手順で詰む可能性があるので、以降は通常の解図と検討を行います。また、作意不詰で作品自体も不詰の場合、作意の末端に近い不詰箇所を表示します。(※1.86版より追加)

手順の指定は kif 形式で行います。(今のところ ki2 形式や他の棋譜形式には対応していません)以下のように手順を入力した kif ファイルを用意し、本プログラムを起動します。



すると、解図の前に入力手順を利用するかどうかの確認が行われます。

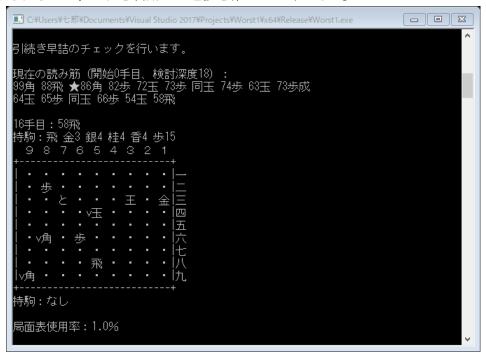


ここで「Y」または「y」を入力してください。 以降は変化の確認と検討が行われます。 「手順指定」の主な用途としては以下を想定しています。

- ・検討時間の短縮(作品にある程度の自信のある場合)
- ・変化の読み抜けがないことの確認
- ・ 手順の清書用
- ・非限定がある場合に特定の手順を検討させたい 等々

【補足1】手順指定時の進捗確認

手順指定を行った場合は、「D」キーによる途中経過表示の時、指定手順と違っている手に「★」マークが付加されるので、これを利用して進捗を確かめてください。



例えば上図では、早詰チェックの工程で、3 手目 85 角の代わりに 86 角を読んでいます。 手順指定がある場合、検討は作意の末端から行われ、初形に遡っていくので、上図は5 手目以降で早詰が見

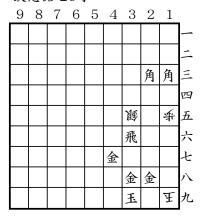
つからなかったことを示しています。

【補足2】作意不詰の場合

入力された手順に不詰の変化がある場合、本プログラムは指定以外の手順で詰むかどうか調べます。 もし詰めば通常通り解を出力しますが、不詰の場合は指定手順の一番末端に近い不詰箇所を表示します。

[不**詰作の例1**] 森茂作、最悪詰 29 手 (詰将棋パラダイス 2001 年 9 月)

森茂作 最悪詰 29手



持駒 なし (詰将棋パラダイス,2001年9月,不詰) 上図の作意は以下の通りですが、19手目28銀のところ、46角成で不詰です。

- 18 金 同玉 28 金 19 玉 18 金 同玉 38 飛 19 玉 18 飛 同玉
- 45 角成 36 銀 同馬 19 玉 18 馬 同玉 29 銀 19 玉 28 銀 18 玉
- 19銀 同玉 46 角生 28 香 同角 18 玉 19 香 27 玉 37 金 まで 29 手

実際に作意手順を入力した.kif ファイルを用意し、本プログラムで解かせてみましょう。



すると、以下のように表示されます。



「作意 19 手目 46 角成(13)で不詰」と表示されていますね。

【補足3】普通詰将棋で不詰を確認する場合の注意点

手順指定による不詰箇所の検出は最悪詰以外のルールでも有効です。

ただし、これを普通詰将棋の不詰チェックに使う場合、少し注意が必要です。

Worst1.exe で取り扱うのはあくまで「最善詰」なので、無駄合いによって伸びる手数の考慮をしないと、手数不足による不詰を検出してしまいます。

また、不詰の変化の意味が単なる「手数稼ぎ」の場合は、検出された不詰箇所以外に「真の不詰」が隠れていることがあります。

具体的な例で説明しましょう。

[不詰作の例2] 宮越宗太郎作「刻」、215 手(将棋世界 1961 年 1 月)

宮越宗太郎 作 「刻」

詰将棋 215手

ス

歩

9 8 7 6 5 4 3 垦 ح 金融金 香 金 香 豣 四 步|爭|桂|爭|步 爭 五 华 # 六 ス 角 ス \boldsymbol{z} と 龍 ス Ŧ ス

持駒 金 (将棋世界,1961年1月,不詰)

角

これを本プログラムで解いてみます。

丰

步九



「作意 162 手目 25 玉(14)で不詰」と出ていますね。 でも、これではなぜこの手で不詰なのか分かりません。ちょっとその局面を見てみましょう。

9	8	7	6	5	4	3	2	1	_
と				季				と	—
			委	餓	委				二
			香	金	香		Ŧ		Ξ
				豣					四
		步	#	桂	华	步			五
				#					六
				角				龍	セ
									八
歩				角				步	九

持駒 桂 歩

(162手目23玉の局面)

作意は 162 手目 23 玉です。そして、ここから「12 龍 24 玉 16 桂…」と進みます。

162 手目 25 玉としても「26 龍 14 玉 15 龍 23 玉」以下作意に合流できるので、これでは「手数稼ぎ」があることしか分かりません。

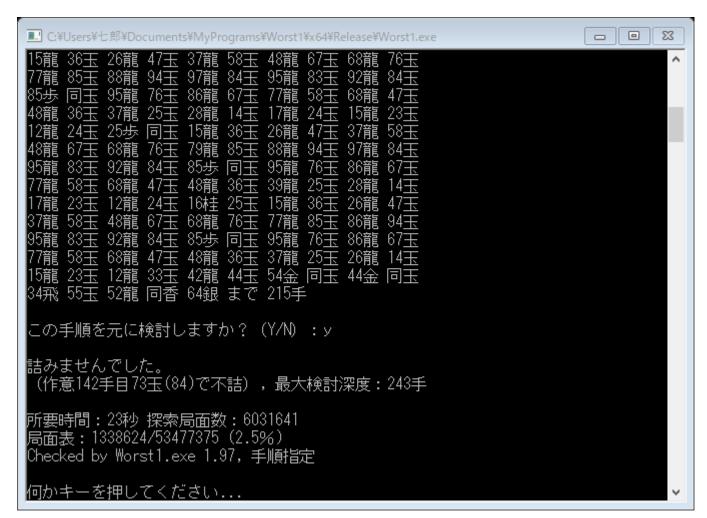
こんな場合に、真の「不詰」を検出する方法があります。

手数を作意ちょうどにするのではなく、余裕を持って指定するのです。

「詰将棋情報」の「手数」欄に書く手数を「215」から「999」に変えて、棋譜を保存します。

詰将棋情報		X
作品番号:		OK
作品名:	刻~	キャンセル
作者:	宮越宗太郎 ~	
発表誌:	将棋世界 ~	
発表年月:	1961年1月 ~	
出典:	~	
手数:	999 ~	
完全性:	不詰~	
分類:	詰将棋 ~	
受賞:	~	
備考:	メモリ1GB ~	

本プログラムで解析すると、今度は「作意 142 手目 73 玉(84)で不詰」と出ました。



実際に作意の142手目と比べてみましょう。

9	8	7	6	5	4	3	2	1	
と				星				と	—
			委	穣	委				_
	Ŧ		香	金	香				ニ
				豣					四
龍		步	*	桂	华	步			五
				#					六
				角					セ
									八
歩				角		丰		步	九

持駒 歩2 (142手目83玉の局面)

作意 142 手目は83 玉でした。確かにここで73 玉とされたら詰みませんね。

以上のように、本プログラムで普通詰将棋を検討し、手順指定で不詰箇所を特定したい場合は、「手数」を大きめに指定することをお勧めします。

2. 詰将棋情報の設定内容

kif ファ	イルの「詰将棋情報」には次の情報を設定します。
項目	説明
手数	手数の上限を指定します。1から2999手が指定可能です。
	最悪詰、最善詰では偶数手のとき受先形式、(最悪・最善) 自玉詰系ルールでは奇数手のと
	き受先形式となります。
分類	ルール名と条件を指定します。ルールは以下の8つが指定可能です。
	「最悪詰」「最悪自玉詰」「最悪自玉ステイルメイト」
	「最善詰」「最善自玉詰」「最善自玉ステイルメイト」「最善勝」「最善ステイルメイト」
	省略時は「最善詰」と解釈されます。※1.97版から無指定時のルールを最善詰に変更
	各ルールには次行に示す条件を付加することができます。
	【最悪詰】
	攻方は受方玉をなるべく詰めないよう王手を掛け、受方はなるべく早く詰むよう応じる
	【最悪自玉詰】
	・ 攻方は攻方玉がなるべく詰まないよう受方玉に王手を掛け、受方はなるべく早く攻方玉
	が詰むよう応じる
	【最悪自玉ステイルメイト】
	攻方は自分がステイルメイトにならないよう受方玉に王手を掛け、受方はなるべく早く
	攻方がステイルメイトになるよう応じる
	【最善詰】※注
	攻方は受方玉がなるべく早く詰むよう王手を掛け、受方はなるべく詰まないよう応じる
	【最善自玉詰】
	攻方は攻方玉がなるべく早く詰むよう受方玉に王手を掛け、受方はなるべく攻方玉が詰
	まないよう応じる
	【最善自玉ステイルメイト】
	攻方は自分がなるべく早くステイルメイトになるよう受方玉に王手を掛け、受方はなる
	べく攻方がステイルメイトにならないよう応じる
	【(非王手可)最善ステイルメイト】
	攻方は受方がなるべく早くステイルメイトになるよう、受方はなるべくステイルメイト
	にならないよう応じる。王手義務なし。「非王手可」は省略可能(自動的に付加)。
	【最善勝】
	攻方はなるべく早く勝つよう着手し、受方はなるべくそれを避けるようよう応じる。
	「勝」は「詰」または「ステイルメイト」を達成すること。攻方王手義務はなし。
	ルール名には異称の用意されているものもあります。
	例えば「最悪自玉詰」は「最悪自殺詰」と指定することも可能です。
	ただし、この場合も出力は「最悪自玉詰」に変更されます。
	※注:「最善詰」は普通詰将棋と同種のものですが、細則は省き抽象的に定義されています。
	無駄合概念はなく、駒が余るかどうかで手順に優劣を付けることもしません。

長手数の余詰は無視します。他の「最善」系のルールも同様です。

条件 ※条件について:各条件の意味は以下の通り。

【禁欲】駒を取らない手を優先して着手を選ぶ

【強欲】駒を取る手を優先して着手を選ぶ

【打歩】打歩詰以外の詰を失敗とする(単純打歩)※1.97版で追加

【マキシ】受方は最長距離の応手を選ぶ。距離は将棋盤を9×9の正方格子とみなした時のユークリッド距離とする。駒を打つ手は距離1と定義する。

【長躯】受方は最長距離の応手を選ぶ。距離は将棋盤を9×9の正方格子とみなした時のチェビシェフ距離とする。駒を打つ手は距離1と定義する。※1.97版で追加

【ミニ】受方は最短距離の応手を選ぶ。距離は将棋盤を 9×9 の正方格子とみなした時のユークリッド距離とする。駒を打つ手は距離1と定義する。%1.97版で追加

【非王手可】攻方王手義務がない(王手をしても良い)。「非連続王手」でも同じ。

「禁欲」・「強欲」・「マキシ」・「長躯」は互いに排他で、同時に指定できません。 条件とルール名の語順は自由です。ただし、出力時には Worst1.exe 内部で決められた順 に従って各名称が並べられます。

備考 | 各種のオプションを指定します。

指定できるオプションは4種あります。複数指定するときはスペースで区切ってください。

[動作に関するオプション]

暫定的な解を求めた後の動作を指定するオプションです。 現在用意されているのは「**解図のみ」「略式検討」「余詰検討」**の3つです。

解図のみ:

「解図のみ」と記入すると、早詰や非限定のチェックを省略します。 変化や紛れを調べるときには便利な機能です。

ただし、変別解が出力される危険もあります。

略式検討:

「略式検討」と記入すると早詰チェックを省略します。

早詰がないことが確実な場合はこれで検討時間を短縮できます。

既に早詰チェックが終わっていて、変別のない手順を得たい場合にも使えます。

余詰検討:(※1.80版より前の「余詰検出」から名称を変更しました)

「余詰検討」は最善系のルールで利用可能なオプションで、攻方の手番で早詰や非限定だけでなく、長手数の余詰も検出する場合に指定します。

単に「余詰検討」とすると、問題の指定手数に応じ 2999 手または 2998 手以内の余詰検 討が行われます。検討する手数を指定する場合は「余詰検討 99」のように、手数を表す 数値を指定します。(※1.91 版から「手数」は初形からの手数の意味に変更しました)

〔求解に関するオプション〕

問題によっては、使用者が条件を外挿した方が速く効率的に解ける場合があります。 この種のオプションで現在用意しているのは**「持駒節約」「対称性検出」**の2つです。

持駒筋約:

このオプションは盤面配置が同一で、手数が同じか大きく、持駒が真に少ない場合、読みを打ち切ります。このオプションを指定できるのは「自玉詰」のみです。誤って指定すると詰むのに不詰となる場合があるので、慎重に使ってください。なお、「禁欲」のつかない「最善詰」は、無条件にこれが適用されます。

対称性検出:

左右対称形で左右対称形の解を無視するオプションです。

途中左右対称もチェックの対象となります。

左右対称形が出現する問題では求解時間を節約できますが、出現しない問題では対称形 検出の時間が無駄になります。

〔出力に関するオプション〕

備考欄に「トレース」と記入すると、探索中の候補手を出力します。 デフォルトのトレース深度は2となっています。

深度を変える場合は「トレース 4」のように、深度を表す数値を添えてください。 また、備考欄に「モニタ」と記入すると、探索中の着手を画面のみに出力します。 使用方法はトレースと同様です。

[メモリに関するオプション]

局面表のサイズを「メモリ 100MB」「メモリ 1GB」のように指定します。

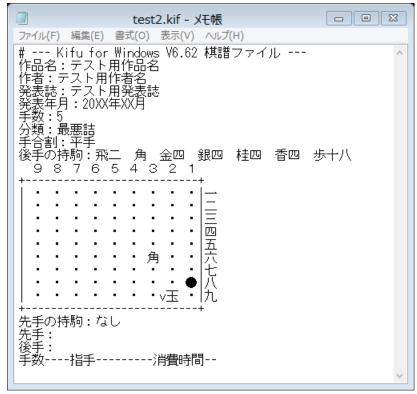
1GB=1024MB です。数値は整数値のみが使えます。GB の単位を使う場合は、メモリ残量にくれぐれも注意してください。無指定時は 24MB と解釈されます。

32bit 版では約1500MB が OS から割り当てて貰えるメモリの上限ですが、64bit 版ではもっと大きな値を指定することができます。解析中は局面表以外にもメモリを使います。マシンの残メモリに注意して、充分に余裕を持った値を指定してください。

3. 特殊な駒、仕様

本プログラムでは f mの「石」と同様、不透過・不可侵のマスを指定することができます。(表記は●) これは Kifuw では扱えませんので直接テキストファイルで指定してください。

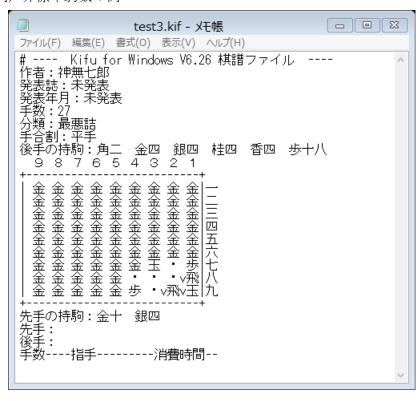
例)石(●で表記)を使った例



また、本プログラムは受方持駒制限や受先形式にも対応しています。

通常より駒数を多く使う問題も設定可能ですが、持駒に関しては九十九枚までという制限があります。枚数は 漢数字で指定し、各持駒の右には全角空白を入れてください。(この指定は kif 形式の書式に準じています。)盤 上の駒数には特段の制限はありません。

例) 非標準駒数の例



4. 動作

このプログラムは3段階の動作をします。

- a. 詰みを見つける
- b. 早詰がないことを確かめる
- c. 非限定・変別を検出する

起動時には実機に搭載されている物理メモリも残量をチェックします。もし不足していた場合、次のようなメッセージが出ますので。何かキーを押してプログラムを終了し、メモリ残量を確認してください。

指定のメモリサイズが実際に使える値より大きくなっています

指定:6144MB, 使用可能:4838MB

何かキーを押してください...

その他、入力データの最低限のチェックが終わると、a.の動作が開始されます。動かし始めたらとにかく解が出るまで待つしかありません。途中で強制終了する場合は、ESC キーの押下で終了してください。

(※1.70 版から中断用のキーを「Ctrl-C」から「ESC」に変更しました)

詰みを発見したときに次のようなメッセージが出てきます。

詰みを見つけました。

暫定的な解を表示します。

このメッセージの後に"暫定的な"解が表示されますが、これは変別を含む可能性があるので、鵜呑みにしてはいけません。ただし、詰みの有無のみを問題にしている場合は、ここで打ち切っても良いでしょう

もし、早詰があれば次のようなメッセージと、検出された早詰解が出ます。

少なくとも2手短い早詰があります。

図や手数指定に誤りがないか確認してください。

暫定的な解を表示します。

この段階で出力された早詰解が最短である保証はないので、プログラムは最短解を求める作業に移ります。 最短解を求めず打ち切る場合は、ESC を入力して中断してください。

なお、指定手数で詰まない場合は、次のメッセージが出ます。

詰みません

この場合、図や手数を再確認してやり直してください。

解が検出されれば、それが指定通りの手数でも早詰解でも a.の段階は完了し、次のメッセージが出ます。

引続き早詰のチェックを行います。

これで b.の早詰チェックの段階に入り、最短解を求める作業に入ります。 早詰チェックが無事終了すると、次のようなメッセージが出てきます。

早詰はありません。変別や非限定のチェックを行います。

このメッセージの後は自動で c.の非限定・変別検出の段階に移行します。

早詰チェックに入る前の段階で既に早詰が見つかっていた場合や、早詰チェックで新たに早詰が見つかった場合は、次のようなメッセージが出ます。(XX は最短解の手数。)

最短解は XX 手でした。

早詰順(暫定解)をもとに、変別や非限定の検討を行いますか? (Y/N):

引続き変別や非限定の検討を行う場合は、「Y」または「v」を入力してください。

ここで「N」または「n」を入力すると検討は終了し、暫定解を出力します。

後で変別や非限定の検討を行う場合は、待避ファイルを作成してくだい。

待避ファイルの作成と、待避ファイルからの再開については「8.中断・再開」をご覧ください。

変別や非限定のチェック中は、暫定解の各手に対して検討を行っている状況が表示されます。 これは、画面のみでファイルには出力されません。

1 手目 16 歩 : 終了

2 手目同玉 : 14 玉 24 玉 25 玉 26 玉 終了

:

まず何手目を検討しているかが表示されます。

変化に対してはまず変別のチェック(暫定解で選んだ手に対し早詰がないことの確認)が行われ、次に暫定解で選択されなかった手が早く詰むこと(同手数変化にならないこと)を確認します。

紛れに対しては暫定解以外の手で非限定(余詰)がないかチェックします。

また、その局面でのすべての選択肢に対する検討が終わると「終了」と表示されます。

変化の検討で変別を検出した際には、次のメッセージが出ます。

変別検出! 代替手を探しています...

この後代替の手が見つかると「手順修正」のメッセージが表示され、修正した手順を元に検討が続行され、「n手目XX:」(nはその時の手数)のメッセージがもう一度出力されます。そのまま最後まで続行すれば、変別のない手順が出力されます。

非限定(余詰)が検出された場合は、画面とファイルの双方に出力されます。 詳しくは「5. 出力ファイル」をご覧ください。

解析が終わった後は次のようなメッセージが出るので Enter キーを押下して終了してください。

所要時間:4秒、探索局面数:106947

局面表: 58091/1048575 (5.5%) Checked by Worst1.exe 1.91

何かキーを押してください...

解析を中断する場合は、ESCキーを押下してください。

解が出るまで待ち切れない時は、頭の手順をカットした図を解かせてみて、徐々に手数を増やすのも良いかもしれません。だんだん解析に時間が掛かるようになるので、最終的な所要時間がある程度予測できます。

また、詰めることはできても、早詰のチェック等に時間が掛かる場合は、備考欄に「解図のみ」と入れて、早詰のチェック以降の工程を省略する方法もあります。この場合、出力される解には変別が混じっている危険もありますが、とりあえず詰むかどうか、どんな手順で詰むかを確認したい場合に有効です。

動作中にメモリが足りなくなった場合は次のようなメッセージが出ることがあります。

メモリが不足気味です(局面表利用率 80.0%) 30.0%の非保護データ(探索レベル 3 以下)を破棄して再利用します

本プログラムでは検討中のデータを局面表に保存し、再利用することによって検討効率を上げていますが、問題が難しいと局面表が満杯に近くなることがあります。そのときに、一部のデータを破棄して、空いた領域を再利用します。

このメッセージが出た時は、できれば多めのメモリを割り当てて再度やり直してください。

このままでも解析は続けられますが、検討の効率は低下します。破棄対象データの探索レベルが一桁くらいなら余裕ですが、二桁を越えるとかなり効率が低下しており、状況が逼迫していると考えられます。

【補足説明】

以前は再利用するデータを選ぶ基準に証明数・反証数を使っていましたが、1.86 版から「探索レベル」の概念を導入し、これを基準に再利用を行う方式に改めています。

「探索レベル」は対象の局面以下の探索木の大きさを、2を底とした対数で表したものです。例えば探索レベルが3の場合、2の3乗、つまり8局面分位の探索を行ったことを示しています。つまり「探索レベル」が1上がるごとに倍のコストを掛けて得られた貴重なデータということになります。従って、なるべく小さな探索レベルのデータから再利用に回す方式になっています。

再利用には「詰データは極力保護する」とか「直近のデータはなるべく保護する」などの基準もありますが、 これらはあくまで内部的な処理であり、表には出てきません。

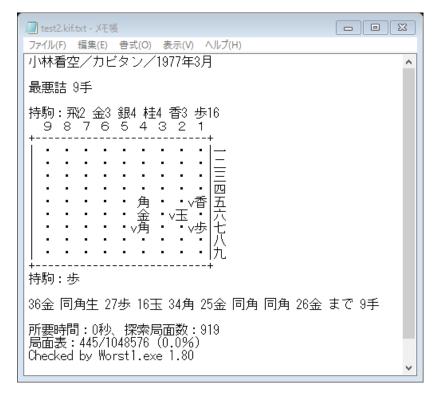
5. 出力ファイル

解析結果は入力ファイルのある場所に「入力ファイル名」+「.txt」の名前で出力されます。従って二重拡張子のファイルになります。また、同じ入力ファイルで連続して解かせると、結果が追記されるのでファイルサイズの増加に注意してください。

出力ファイルの末尾にはプログラムの版数が表示されます。バグなどの報告のときは、この版数も併せて報告してください。

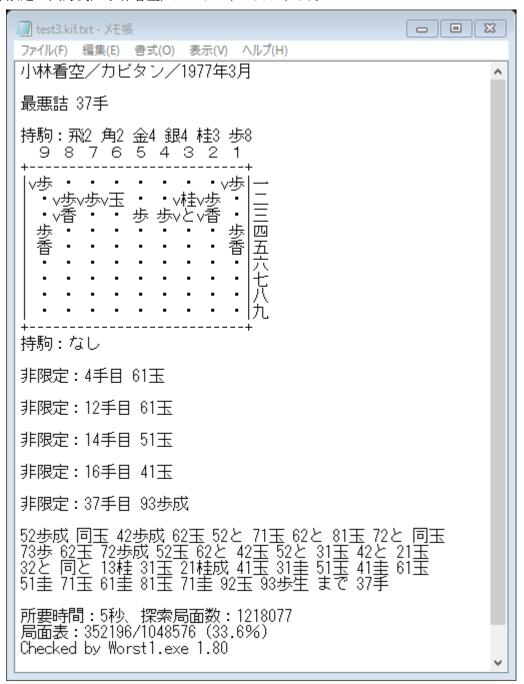
このファイルは fmview で鑑賞可能な形式になっています。

出力例) 小林看空/カピタン/1977年3月



ちなみに、非限定がある場合は、最初に見つかった解とそれに対する非限定が出力されます。 非限定はその存在のみが報告され、手順は示されません。非限定の中に更に非限定があっても無視されます。 解手順より非限定の方が先に出力されますが、気にしないでください。

非限定の出力例) 小林看空/カピタン/1977年3月



もし「詰(あるいはステイルメイト)」を証明する側に非限定があったら「余詰」と表示すべきかもしれませんが、今はどちらの側も作意以外の解は「非限定」としています。

左右対称形からの左右対称解も非限定扱いされますが、それを無視するかどうかは使用者の方で判断してください。無視する場合は、あらかじめ「備考」に「対称性検出」と指定すると、左右対称形からの左右対称解を除外します。

また、「余詰検討」のオプションを使った場合、詰を証明する側の非限定は「余詰」と表示されます。

このとき、既出の局面に戻る余詰は「迂回」と表示されます。(※1.86版より変更)

「迂回」と表示されても、本当の余詰がないことの保証にはなりません。あくまで、チェックの目安程度に 考えてください。

6. 注意 (特に普通詰将棋を解く場合)

(1) 普通の詰将棋と「最善詰」の違い

本プログラムでは「最善語」が機能に含まれ、普通の語将棋もある程度解くことができます。 ただし、「最善語」では「攻方最短・受方最長」以外の規則はすべて除外しています。 具体的に違いが生じるのは以下のような事項です。

- ・無駄合概念はない
- ・駒が余るかどうかで手順に優劣を付けない
- ・長手数の余詰は無視する

このようにしているのは開発の手間を省くためではなく、本プログラムがフェアリー詰将棋用のプログラムだからです。

フェアリーではルール単体を最適化するために細則を設けるより、ルールの応用を重視してルール群全体の 最適化を考えます。

その場合、コアとなる「どのような手を選ぶか」というロジックは可能な限りシンプルでなければなりません。普通の詰将棋の規則をその附則と一緒にフェアリーに導入するのは望ましくないのです。

そのため、本プログラムでも「攻方最短・受方最長」以外の要素を排除した「最善詰」を扱っています。(「最善自玉詰」や「最善自玉ステイルメイト」でも、攻方・受方の役割は異なりますが、同様の扱いです。)

以下に「最善詰」と普通の詰将棋でどのような違いが現れるか、具体的に見てみましょう。

例)

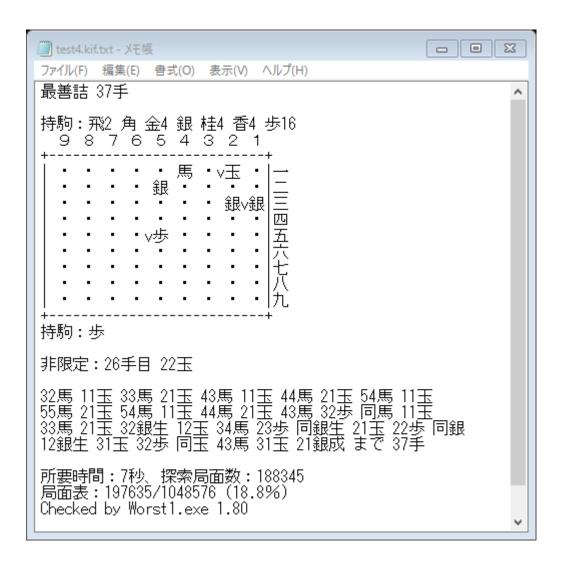
9	8	7	6	5	4	3	2	1	_
					馬		Ŧ		—
				銀] =
							銀	憵	Ξ
									四
				华					五
									五六七
									セ
									八
									九

持駒 歩

これを普通の詰将棋として解くと、以下の35手詰が正解となります。

32 馬 11 玉 33 馬 21 玉 43 馬 11 玉 44 馬 21 玉 54 馬 11 玉 55 馬 21 玉 54 馬 11 玉 44 馬 21 玉 43 馬 11 玉 33 馬 21 玉 32 銀生 12 玉 34 馬 22 玉 23 銀生 31 玉 32 歩 21 玉 22 歩 同銀 12 銀生 32 玉 43 馬 31 玉 21 銀成 まで 35 手

しかし、これを「最善詰」として本プログラムで解くと、以下のようになります。



17 手目「32 歩」は普通の詰将棋では「無駄合」とされる手ですが、「最善詰」ではこれを省略しません。 普通の詰将棋に慣れていると、この種の合駒を除外することに慣れているので、しばしば手数不足の「不詰」 を引き起こします。特に無駄合が変化に隠れている場合に要注意です。

また、26 手目「23 歩」は「同手数駒余り」であり、普通の詰将棋では手数の短い変化と同様に扱われますが、「最善詰」では手数の同じ「22 玉」の変化とまったく同格の扱いです。仮に、選択された解が 26 手目「22 玉」なら 28 手目「31 玉」の非限定も検出されるはずですが、今回はたまたま 26 手目「23 歩」が選択されたため、出力されていません。「非限定」の中の「非限定」は無視されるのです。

本プログラムで出力される「非限定」が普通の詰将棋でも非限定かどうかは、使用者自身が判断する必要があります。

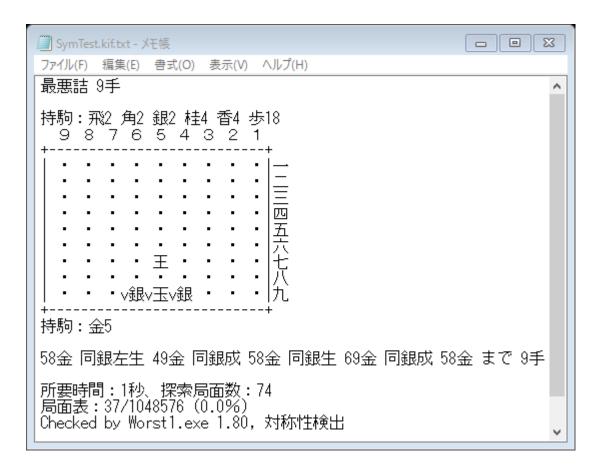
(2) 千日手の扱い

本プログラムは「王手の千日手」を禁手と扱いません。そのため、このルールを適用する作品の解図はできません。

(3) 左右対称の扱い

本プログラムのデフォルトは、左右対称形で手順が左右対称であっても「非限定」として扱います。それが本当に非限定かどうかは使用者が判断してください。

「備考:」欄に「対称性検出」のオプションを指定すると、左右対称形からの左右対称解を無視します。



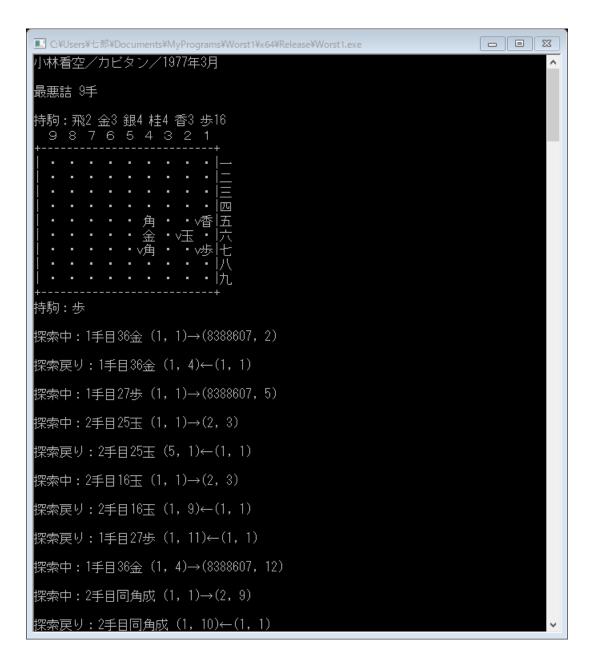
途中から左右対称になる問題でもこのオプションは有効で、うまくいけば検討時間を大幅に短縮することができます。ただし、左右対称形が現れない問題では、このオプションは逆効果で、時間のロスになります。

7. トレース、モニタ機能と本プログラムの動作原理

(1) トレース、モニタ機能について

本プログラムには、プログラムの挙動を知るために「トレース」「モニタ」という2つの機能を備えています。これは出力先が「ファイルと画面の両方」か「画面のみ」かという違いだけで、ほとんど同じ機能です。ただ、「トレース」はしばしば出力ファイルのサイズを膨大にしてしまうので、普段は「モニタ」を使うことを推奨します。

以下では「モニタ」の使用例とその見方について説明します。



初形図の下に「探索中」や「探索戻り」で表示されているのが「モニタ」の内容です。デフォルトではモニタの深度は「2」なので、2手目まで、どのような手を読んでいるかここに示されています。

横に数字の組が表示されていますが、それが何を意味しているか気にする必要はありません。興味のある方は後述の解説をお読みください。

一般の使用者にとって、注目すべきは作意と合致する手の出現頻度です。プログラムが最初のうちは迷走しているように見えても、だんだん作意順に読みが絞られてきたら、正解に近づいていることが推測できます。

[モニタ、トレースについての解説](※興味のある方のみお読みください)

上記の図を例として説明します。

まず1行目「探索中:1手目36金」は、初手36金を読んでいることを表しています。その横に(1,1)という数字の組がありますが、これは36金を指した後の局面に与えられている「証明数」と「反証数」の組を表します。現時点では36金から先はまったく読んでいないので、証明数と反証数は共に「不明」を表す「1」が与えられています。

今、説明なしに「証明数」「反証数」という言葉を使いましたが、証明数は「詰み」(あるいは目的達成)を 証明するのに読む必要のある局面数、反証数は「不詰」(あるいは目的達成不可能)を証明するのに読む必要の ある局面数と思ってください。

次に「 \rightarrow 」の後に(8388607、2)という数字の組が出てきました。これは証明数の閾値、及び反証数の閾値の組です。閾値というのは、この数に達したら読みを打ち切ろうという基準値を表します。つまり証明数の閾値が8388607というのは「36 金とした局面は証明数が8388607になるまで頑張って調べよう」ということを表し、反証数の閾値が2というのは「36 金とした局面は反証数が2になったら早々に出直そう」ということを表しています。最悪詰では攻方は「不詰」を証明する立場ですから、なるべく少ない読みで「不詰」を証明できると都合が良いので、その期待に基づいて、なるべく小さい閾値を反証数に割り当てているのです。ただし、この閾値は固定ではなく、読みの進展に合わせて変化していきます。

次の行は早々に探索から戻ってきたことを示しています。ここで証明数・反証数の組は(1,4)となっています。実際 36 金に対しては「同角成」「同角生」「16 玉」「27 玉」の 4 つの受けがあり、攻方はそのどれもが詰まないことを示さなくてはいけません。仮にすべての受けが反証数 1 の範囲で詰まないとしても、反証数は 4 より小さくはなりません。そして「4」は反証数の閾値として与えた「2」よりも大きかったので、探索を切り上げて戻ってきたのです。

なお、「←」の後に(1,1)という数字の組が出ていますが、これは探索を行う前の証明数・反証数の組の値を表しています。一般には証明数が大きくなり、反証数が小さくなったら「不詰」の可能性が強く、逆に証明数が小さく、反証数が大きくなったら「詰」の可能性が強いわけです。ただし、その先の局面を読み進めると突然その傾向が逆転することがあり、あくまで現時点での詰みやすさ詰みにくさを評価した目安に過ぎません。 三行目は攻方の別の初手が出てきます。今度は27歩を選択し、閾値に(8388607,5)を与えていますね。 先程読んだ36金の反証数が4だったので、それより一つ大きい5を反証数の閾値に設定したわけです。

四行目では 27 歩に対し、25 玉を読んでいます。これも最初は「不明」の局面なので(1,1)の組なのですが、 閾値は (2,3) が与えられています。まずは証明数の閾値がたったの 2 であることに着目してください。最悪 詰では受方は「詰み」を証明する側なので、なるべく証明数を小さく抑えたいわけです。これは最初の 36 金で攻方が反証数の閾値を 2 にしたのと同様の理屈です。

一方、25 玉の反証数の閾値が 3 である理由は少し考えないと分からないでしょう。まず、初手 27 歩とした局面では反証数の閾値に既に 5 が与えられています。一方、この局面には今回選択する「25 玉」の他に「16 玉」と「37 玉」という 2 つの応手があります。それぞれの反証数が 1 だとしても、合計は 2。従って「25 玉」の反証数が 3 に達したら、上限の 5 に達してしまいます。というわけで、25 玉に対する反証数の閾値は 3 より大きくできません。その結果、閾値の組は(2, 3)となっています。

更に次の行は「探索戻り」となっています。これは先ほどの 25 玉を指した局面が、期待に反して「証明数 2」「反証数 3」の範囲内で解けなかったため、これを指す前の局面に戻ってきたことを示しています。そして、戻ってきたときの数字の組は(5, 1)に更新されています。証明数と反証数のどちらかが閾値に達したら戻ってくるわけですね。

実際 25 玉に対しては「26 歩」「34 角」「36 角」「35 金」「36 金」の 5 つの王手があります。だから、この局面の詰・不詰について結論を出すには、5 つの王手すべてについて「詰」を証明するか、どれか 1 つについて「不詰」を証明しないといけません。証明数と反証数の組が(5, 1)になったのはこのためです。

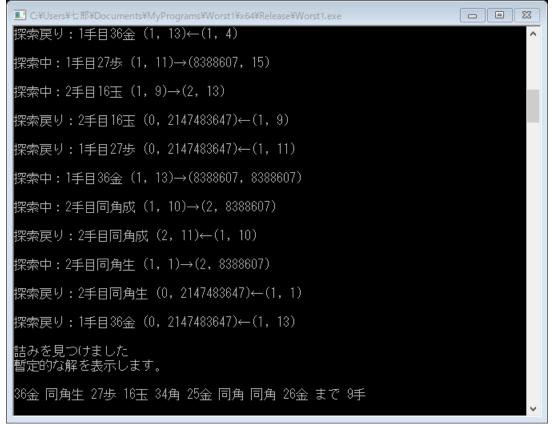
少し飛ばして 8 行目。「25 玉」も「16 玉」も打ち切りになって、この時点で「37 玉」を読むまでもなく、「27 歩」の読みを打ち切っています。これは「16 玉」の反証数が 9 で、「27 歩」に与えられた反証数の閾値である 5 を上回ってしまったからです。

そして 9 行目、再び初手 36 金を読んでいます。ただし、今度は反証数の閾値を 12 に設定し直しています。 このように局面を前へ後ろへと移動しながら、証明数や反証数を徐々に変化させ探索範囲を広げていきます。 この証明数・反証数や各閾値の更新の仕方は「df-pn」*1ではなく、そのアレンジ版「WPNS」*2を利用しています。合流やループの多い自玉詰系や最悪詰系の詰将棋にはこちらの方が向いているとの判断です。

(※1.97 版より df-pn が使えそうな場合にはそれを優先するよう変更しました。そのため証明数・反証数の値は以前の版より大きめになります。)

また、難問を解く場合、細かく閾値を上げていくのではなく、最初からある程度大きめの閾値を与えるという小細工 $(1+\epsilon \text{Trick})^{*3}$ なども使用しています。

さて探索が始まる方は見たので、今度は探索が終わるところを見てみましょう。



上の4行目に注目してください。「探索戻り:2手目16玉 (0,2147483647)」となっています。

証明数が 0 — これは実際に証明ができたことを示します。(このとき反証数は理論上 ∞ (無限大)を割り当てるのですが、普通のプログラミング言語にはそんな数はないので、代わりにできるだけ大きい数を使っています。)それによって必然的に、初手 27 歩も詰みとなり、5 行目「探索戻り: 1 手目 27 歩 (0,2147483647)」が表示されています。実際初手 <math>27 歩は以下 16 玉 34 角 25 金 同角 同角 26 金 まで 7 手で詰むので、確かにこの結果は正しいのです。

更に 11 行目、今度は初手 36 金も(0, 2147483647)となりました。初手 36 金も初手 27 歩も詰みだと分かったので、初形が詰みだということも判明しました。解図終了です。

不詰の場合は数字の組が(2147483647,0)と逆になるので、作意の手順に(0,2147483647)が現れ、詰まないはずの手順に(2147483647,0)が現れれば、順調に解図が進んでいると推測することができるわけです。逆に、このような極端な数字が出ないうちは、プログラムが苦戦しているのだと思ってください。

また、証明数や反証数の閾値の最大値は 8388607 に設定しています。従って(8388607, 8388607)という数字の組が出始めたら、そこで読んでいる手以外は詰か不詰の結論が出たと思って良いでしょう。これも解図進行の一つの目安になります。ただし、早詰チェック以降の工程では、作意以外の手や、作意通りの手でも与えられた手数が短いと詰まないはずなので、ある程度調べた段階で、いきなり閾値を(8388607, 8388607)に引き上げることがあります。

[参考文献]

- ※1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」、VoL43, No6, 情報処理学会論 文誌, June, 2002 (http://ci.nii.ac.jp/naid/110002726401)
- ※2 橋本 剛、上田 徹、橋本隼一「オセロ求解へ向けた取り組み」、組合せゲーム・パズル ミニプロジェクト、 第3回ミニ研究集会 2008 年 3 月 7 日(http://www.alg.cei.uec.ac.jp/itohiro/Games/Game080307.html#an chor3)
- **3Jakub Pawlewicz, Łukasz Lew 「Improving Depth-first PN-Search:1+eTrick」、5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. (http://www.mimuw.edu.pl/~pan/papers/lm-pns.pdf)

(2) モニタの表示・非表示の切り替えと深度の変更

以前は「モニタ」は早詰チェック以降の工程では出さない仕様でした。これは暫定解、早詰、非限定といった重要な情報が、画面外に流されないための仕組みだったのですが、検討が長時間に及び、挙動を確認したい場合には不便でした。

そこで 1.80 版からはモニタを表示するかどうかを「M」または「m」キーの押下で切り替えられるようにしています。モニタをしていない状態で「M」キーを押下すると以下の表示が出て、モニタが開始されます。

モニタ開始:2手

もう一度「M」キーを押下すると、以下の表示が出て、モニタが停止します。

モニタ停止

モニタ深度の初期値は「2」なので、プログラムが深い探索をしている場合は、何も表示されません。そこで、モニタ深度の変更を行う機能を使います。「+」キーの押下で深度を増やし、「-」キーの押下で深度を減らします。

 モニタ開始:2手
 (ここで + キーを押下)

 モニタ深度:3手
 (ここで + キーを押下)

 モニタ深度:4手
 (ここで - キーを押下)

 モニタ深度:4手
 (ここで - キーを押下)

 モニタ深度:3手

なお、この「深度」は開始手数からの相対的な手数を表します。

従って変別や非限定を検討している段階では、検討中の手数を足した手数の状況が表示されます。 例えば 5 手目の非限定の検討中にモニタ深度を 2 とすれば、5 手目から 7 手目の探索の状況が表示されます。

プログラムが現在どの付近を探索しているかは「D」キーの押下で途中経過表示することにより分かるので、 それを参考に、モニタ深度を調整してください。

また、早詰チェック終了時、非限定・変別の各着手のチェックの終了時、画面表示が流れないようモニタ表示は自動的にオフになります。モニタ深度は維持されます。

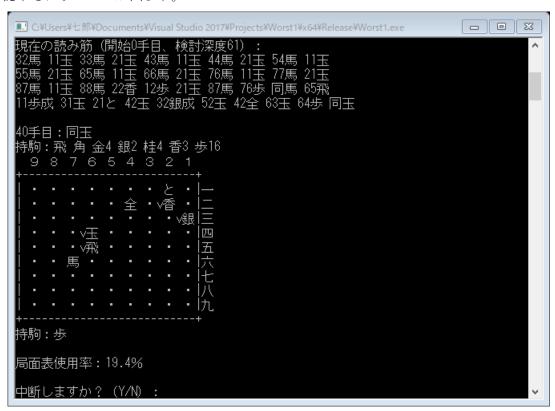
8. 中断・再開・途中経過の表示

本プログラムは長時間の検討に資するため、検討を途中で中断する機能と、中断した状態から再開する機能 を備えています。

【注意】1.70版から中断用のキーの割り当てを Ctrl-C から ESC に変更しました。 Ctrl-C を押下しても「本プログラムの終了には Ctrl-C ではなく ESC を使ってください」のメッセージが出るだけです。

(1) 中断

本プログラムは探索中に ESC キーの押下で、プログラムを一旦停止させることができます。プログラムが ESC キーの押下を検出すると、現在探索中の手順と局面、及び局面表使用率が表示され、中断するかどうか確認するメッセージが出ます。



ここで「Y」または「y」を入力すると解図を打ち切り、「中断します。」というメッセージを表示します。 逆に「Y」や「y」以外の文字を入力すると解図は続行されます。 つまり、途中経過を観察する目的でこの機能を利用できます。

中断が選択されると待避ファイルを作成するかどうか問い合わせるメッセージが出ます。

中断します。

現在の状態を保存しますか? (Y/N):

このメッセージに対して「Y」または「y」を入力すると待避ファイルの作成を開始し、以下のメッセージが出力されます。

待避ファイル作成中... しばらくお待ちください

待避ファイルには現在の問題・探索中の手順・局面表のデータ等が保存されます。このデータは元のファイルがあったのと同じ場所に、元の名前に「.w1s」という拡張子を付けて保管されます。

【注意】待避ファイルのサイズは使用メモリのサイズとほぼ同じ大きさになります。 使用メモリに大きな値を割り当てていた場合は、ディスクの残り容量にも気を付けてください。

なお、ESC キー押下が、暫定解を求めた後だった場合、待避ファイルを作るかどうかの前に、暫定解をファイルに残しておくかどうか問い合わせるメッセージが表示されます。

暫定的な解を出力しますか? (Y/N):

ここで「Y」または「y」を入力すると出力ファイルに暫定解が出力されます。

暫定解を残しておきたい場合は、「Y」または「y」を入力してください。

何度も中断・再開して同じ解の重複出力を避ける場合や、最終的な解のみを残す場合はここで「Y」や「y」 以外の文字を入力してください。

待避ファイルの作成が完了すると、以下のようなメッセージが表示されます。

所要時間:6秒、探索局面数:991034 局面表:131839/1048575 (12.6%) Checked by Worst1.exe 1.91

(中断:求解中)

何かキーを押してください...

解析終了時とほぼ同じメッセージですが、「Checked by Worst1.exe 1.XX」の後に「中断」の文字と、どの工程で中断されたかの情報が付加されるところが異なります。Enter キー等を押下して終了してください。

(2) 再開

再開時に特別な操作は不要です。最初から解析を開始するのと同じ操作をしてください。

このとき、元のファイルの名前を元に待避ファイルを探しに行きます。

ですから、<mark>ファイル名の変更や移動はしないでください。</mark>

また、待避ファイルを作成したときと問題の設定(ルールや手数など)や、オプション(使用メモリの設定など)が異なっていた場合は、再開ができません。

待避ファイルが見つかったら以下のメッセージが出ます。

待避ファイルがあります。これを使って再開しますか? (Y/N)

ここで「Y」や「y」を入力すると、プログラムは「待避ファイルから再開します。」というメッセージを表示し、待避ファイルに保存されたデータを元に解析を再開します。逆に「N」や「n」を入力すると、待避ファイルは無視して最初から解析を行います。

待避ファイルからのデータの読み込みが終わると、問題図・中断時の手順・中断時の局面が表示されます。 ただし待避ファイルを作成したのが早詰チェックの段階の場合は、再開時に手数が元の手数より2手少なく 表示され、手数の横に「早詰チェック」の文字が付加されます。

待避ファイルを作成したのが変別・非限定チェックの段階の場合は、再開時に何手目のどの手から再開する か示されます。検討が済んだ手順の分については「検討済」と表示され、検討をスキップします。

解析が終了したとき待避ファイルを削除するかどうか問い合わせるメッセージが出ます。

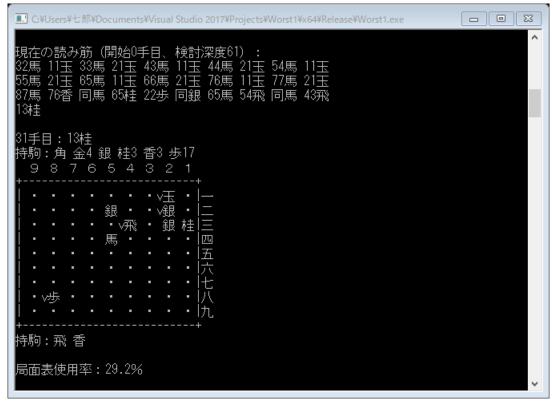
待避ファイルを削除しますか? (Y/N):

ここで「Y」や「y」を入力すると、待避ファイルを削除します。

「Y」や「y」以外の文字を入力した場合、待避ファイルは自動で削除されません。 待避ファイルを残した後で不要になった場合は、エクスプローラなどから手動で削除してください。

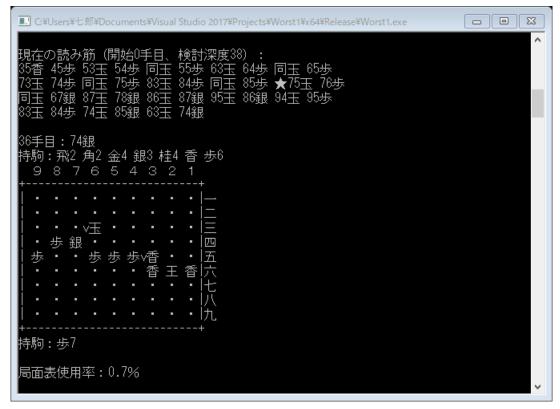
(3) 途中経過の表示

解析中に「D」または「d」のキーを押下すると、その時点で読んでいる手順と図面が表示されます。



中断の場合、再開させるまで処理は止まりますが、この方法では途中経過を表示した後、そのまま処理が継続されます。

「手順指定」での検討の場合、指定手順と違っている手に「★」マークが付加されます。



このマークが変化に付いている場合は、順調に作意通り進んでいることを意味し、作意では詰まないはずの

紛れに付いている場合は、どこかで詰まない変化があったために別の手順を読んでいることを意味します。 作意に問題がない場合、「★」マークは詰上りから初形まで遡るように移動するはずです。

早詰チェック以降の工程にすぐ進まない場合、難しい変化を読んでいるのか、作意自体が誤っているのか確認するため、途中経過表示を利用して進捗を確かめてください。

また、早詰チェックの段階では、どの手でも詰まないはずなので、必ず紛れの手に「★」マークが付きます。 この時も最終手から初手に遡るように「★」マークが移動するので、進捗の確認に使うことができます。

9. 動作に関するオプションについての補足

本プログラムは解図から検討までの作業を連続して行いますが、解図後の変別の除去や完全性の検討に関するオプション動作を kif ファイルの「備考」欄に指定して変えることができます。現在使用できるのは「解図のみ」「略式検討」「余詰検討」の3つです。

このうち、「解図のみ」「略式検討」は検討作業を省略または簡素化するオプションです。本プログラムは性能の限界などもあり、非常に時間が掛かることがあります。また、詰みを見つけるのは短時間でも、完全性を確認するのは多大な時間が掛かることがしばしばあります。そんな時のために用意したオプションが「解図のみ」と「略式検討」です。

また、「余詰検討」は長手数の余詰がないか検討するオプションです。「最善詰」が普及していない現状では、フェアリーでも普通詰将棋と同様、長手数の余詰があると不完全作扱いとなるルールで創作・解答募集が行われる場合が多いので、このオプションを追加しました。

(1) 解図のみ

「解図のみ」は詰むことを確認したら、ただちに暫定解を出力し、プログラムを終わらせるオプションです。 暫定解には変別が含まれる可能性がありますが、詰むか詰まないか確認するだけなら役に立ちます。

(2) 略式検討

「略式検討」と記入すると、早詰チェックを省略して変別・非限定のチェックを行います。早詰がないことが確実な場合や、変別を除いた手順を得たい場合に使用を検討してください。早詰チェックが終わらないような問題を略式検討で解いても、ちゃんとした検討にはならないのですが、どこで検討作業が行き詰っているかを確かめるのには役立つ可能性もあります。

画面表示は「早詰はありません。変別や非限定のチェックを行います」というメッセージが出ないだけで、 後は通常の表示と同じです。

(3) 余詰検討

最善系のルールで「余詰検討」と記入すると、攻方の手番で作意より長手数の余詰がないかチェックします。 普通詰将棋のルールに準じた作品を検討するときに利用してください。余詰が見つかると、他の非限定などと 同様に、解手順の前に「余詰:」の文字列と共に最初の1手が出力されます。手順は出力されませんので、詳 しく調べる場合は別途余詰の局面を作って本プログラムに解かせてください。

余詰検討の手数は実際には無制限ではなく、2999 手または 2998 手の範囲で行われます。(どちらの手数になるかは元の問題の手数の偶奇によります)。この手数を変更したい場合は「余詰検討 100」のように、手数を示す数値を「余詰検討」の文字列の後に続けて記入します。※1.91版より手数は、初形からの手数になりました。

このオプションを指定して検討を行ったとき、終了時には以下のように出力されます。

Checked by Worst1.exe 1.94, 検討深度(実/指定) 61/2999

上の表示は上限手数が 2999 手で、実際には 61 手以内で検討が完了したことを示します。

なお、本プログラムの余詰検討はあくまで「最善」ルールで解いた解に対して行うものです。無駄合や同手数駒余りなどは「最善」ルールでは考慮されないので、普通詰将棋としての余詰とは一致しない場合があります。手順前後も「余詰」とされるので、普通の詰将棋の感覚からは違和感があると思いますが、あくまで「最善」ルールの立場から検討結果が表示されるので、それはそういうものだと思ってください。既出の局面に戻る余詰は「迂回」と表示しますが、既出局面に合流するまでに別の余詰が隠れていないことは保証しません。

【注意】1.94 版から迂回チェックの手数制限を排したので、迂回手順中の余詰には一層気を付けてください。

10. 性能の限界

本プログラムの使用は忍耐力との戦いです。

詰みを見つけるのはまあまあ早いのですが、早詰チェック以降は大抵多大な時間が必要です。

ルール固有の高速化ロジック (例えば「最善語」における局面間の優越関係の利用) は入れていません。 特定のルールで有効なロジックが別のルールではまったく役に立たなかったり、あるいは有害な影響を及ぼ したりすることがあるからです。今後も高速化は検討していきますが、それはルールに依存しない範囲で行う か、オプションとして提供したいと思います。

どの程度の手数が扱えるかは作品の性質によるので一概には言えませんが、だいたい 50 手が実用上の限界と思ってください。短手数でも多大な時間が掛かるもの(裸玉や合駒物)もあれば、超長手数でも実用的な時間に収まるものもあります。

解図・検討に掛かる時間を前もって予測することはできませんが、モニタ機能や中断機能を活用し、プログラムの挙動から解けそうかどうか判断してください。1.50版からfmに倣って、中断・再開機能を実装したので、長時間の検討が可能となっています。性能の不足は大量の時間を投入することで補ってください。

11. 連絡先

本プログラムについて不具合の指摘、要望などがありましたら神無七郎(k7ro.ts@gmail.com)までご連絡ください。掲示板(http://k7ro.sakura.ne.jp/cgi/upload/fairytalks.cgi)でお知らせいただいても結構です。

99. 更新履歴

2021 00 22	1.05	、 タル) - 「 、 「 日 町 .
2021.09.28	1.97	▶条件に「ミニ」「長駆」(チェビシェフ距離を使ったマキシ)「打歩」(単純打歩)
		の追加。
		▶ルール無指定時の解釈を「最悪詰」から「最善詰」に変更
		▶早詰チェック以降、各着手の検討終了時にモニタ表示を自動で終了
		▶解析時間表示を時分秒に変更
2021.02.23	1.94	▶空き物理メモリのチェックを追加
		▶「余詰検討」指定時に実際の最大検討深度を表示
		▶迂回チェックの手数制限を撤廃
		▶その他不具合修正等
2020.05.20	1.91	▶マキシ条件の追加
		▶ ウィンドウの「閉じる」ボタンの無効化
		▶余詰検討の手数を「初形からの手数」の意味に変更
		▶ 初形のチェック強化
		▶探索戻り時、モニタ・トレースに前回の証明数・反証数の情報を追加
		▶その他不具合修正等
2018.05.01	1.86	▶ 持駒枚数上限を 99 枚に拡張。
		▶手順指定で不詰時、作意不詰箇所の表示。
		▶局面表データの再利用基準を探索木の大きさのレベルに変更。
		▶中断時に進捗に関する情報を出力。
		▶余詰のうち4手以内に既出の局面に戻る可能性のある解を「迂回」と表示
		▶非限定検出時の表示色を変更、変別・非限定チェック時の桁揃え。
		▶その他不具合修正等
2017.02.04	1.80	▶取扱いルールに「非連続王手(非王手可)」及び「勝」を追加
		▶オプション「余詰検出」を「余詰検討」に変更。
		▶M キーの押下でモニタを ON・OFF する機能の追加。+-によるモニタ深度の調
		整の追加。途中経過表示に局面表使用率の情報を追加。
		▶早詰変別チェック時の桁揃え処理追加、手数上限の拡張
		▶手順指定の追加(kif 形式のみ)
		▶取扱う手数を 1~2999 手に変更
		▶その他不具合修正
2016.02.25	1.74	▶メモリのサイズ指定を選択から任意に
L	L	1

		× 冷山辺辺のまご吐は図ますまます。
		▶途中経過の表示時に図面も表示する
		▶取扱う手数を 2~2000 手から 1~1999 手に変更
		▶持駒節約オプションを「強欲」でも指定可に
		▶その他不具合修正など
2015.12.27	1.70	▶中断のトリガーを Ctrl-C から ESC に変更
		▶D または d による途中経過の表示機能の追加
		▶局面表再利用方式の変更
		▶対称性検出処理の漏れの修正
		▶再開時の手順表示の追加
		▶その他不具合修正など
2015.11.05	1.67	▶「対称性検出」「持駒節約」のオプション追加
		▶詰専用局面表の廃止と局面表の統合
		▶モニタ・トレースの出力形式の変更(指定深度からの戻りも表示)
		▶その他不具合修正、改良など
2015.06.25	1.61	▶「余詰検出」のオプション追加
		▶閾値上限の計算を厳格化
		▶暫定解を求める処理の改良、変別チェック処理の不具合の修正
2015.04.27	1.59	▶非限定・変別の中断からの再開で、再開手が読み飛ばされる不具合の修正。
		▶メモリサイズの単位に GB を追加
		▶禁欲、強欲条件の追加
2015.03.01	1.54	▶ 単玉ステイルメイトの判定に関する不具合の修正
2015.02.23	1.53	▶局面表再利用方式の改善
		▶手数が余分に指定されたとき、不適切な暫定解が出力されることがある不具合
		の修正
2014.12.23	1.50	▶中断・再開機能の追加
		▶ 手数の上限を 1999 手に拡張
		▶ 早詰検出時、最短解を求めるよう変更
		▶ファイル出力を追記式に変更
		▶不詰専用局面表の廃止
		▶開発環境を Visual Studio Community 2013 に移行
2014.10.13	1.43	▶一部、詰むはずの作が詰まなくなっていた不具合の修正
		▶収束4手目の変別チェックを行わない不具合の修正
		▶ 非限定・変別チェックの画面出力書式の変更
2014.10.05	1.40	▶取り扱うルールに最悪自玉ステイルメイト、最善詰、最善自玉詰、最善自玉ス
2011110100	1.10	テイルメイトを追加
		▶中断時処理の改善
		▶伝統的表記の導入
		▶トレース・モニタの仕様変更
		▶ 開発環境を Express 2012 for Windows Desktop に移行
		▶ 64bit 対応版の導入、Windows 2000 以前のサポートの停止
2010.08.18	1.20	▶ 取り扱うルールに最悪自玉詰(最悪自殺詰)を追加
2010.00.10	1.20	→手順出力に手数追加
2009.02.12	1.12	▶ ビルドオプションを変更(MFC をスタティックリンクに。速度重視の最適化。)
2008.07.19	1.12	→ 開発環境を VC++2008 Express Edition に移行
2000.07.13	1.10	▶ 手数を余分に指定すると不詰になるバグの修正
		プーチ数を示力に指定すると不能になるバクの修正プーテンデーを表合があるバグの修正
		▶ DFS オプションの廃止
2007.07.05	1.07	▶トレース表示の簡略化 等々
2007.07.05	1.07	▶ モニタ機能の追加、処理の一部の改善 ▶ 出力スッカージの微調整
2007.06.13	1.06	▶ 出力メッセージの微調整
2007.06.05	1.05	▶ 早詰チェックを省略する機能を追加 ▶ 不計の割別はなかれるがどの修工
		▶ 不詰の誤判定をするバグの修正

2007.05.16	1.04	▶途中経過表示に候補手を追加
		▶変別・非限定チェックのトレースをオフに戻した
2007.05.14	1.03	▶変別・非限定チェック時の経過表示機能の追加
		▶ 強制終了時のファイル出力
		▶ 非効率な処理の一部を改善
2007.04.28	1.02	▶誤った非限定を検出するバグの修正
2007.04.25	1.01	▶成れる位置から角を打たないバグの修正
2007.04.01		▶ 初版公開

以上