# AN IMPROVER OF CHESS PROBLEMS

YAAKOV HACOHEN-KERNER

Department of Computer Science, Jerusalem College of Technology (Machon Lev), Jerusalem, Israel

NAHUM COHEN
EREZ SHASHA

Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan, Israel

Computer chess programs achieve outstanding results at playing chess. However, no existing program can compose adequate chess problems. In this paper, we present a model that is capable of improving the quality of some of the existing chess problems. We have formalized a major part of the knowledge needed for evaluating the quality of chess problems. In the model, we attempt to improve a given problem by a series of meaningful chess transformations, using a hill-climbing search, while satisfying several criteria at each step. This model has been implemented in a working system called Improver of Chess Problems (ICP). The results of the experiment we carried out show that the majority of the problems examined were optimal. However, the software has improved almost one-third of the tested problems—most of them needing only slight changes. General lessons learned from this research may be useful in other composition domains.

Over the years, chess has proven to be a very interesting and attractive domain, while also proving fertile ground for techniques and ideas that

Address correspondence to Yaakov Hacohen-Kerner, 21 Havaad Haleumi St., P.O.B. 16031, 91160 Jerusalem, Israel. E-mail: kerner@avoda.jct.ac.il

Cybernetics and Systems: An International Journal, 30:441–465, 1999

0196-9722/99 $12.00 + .00

have later been used in various domains of computer science in general, and of Artificial Intelligence (AI) in particular. Computer chess programs attain the highest level at chess playing—the grandmaster rank.

However, to the best of our knowledge, there is no program able to compose adequate chess mate problems. In a close domain chess endgames, Grandmaster Nunn (1993) published some nice endgame studies. He used computer-generated endgame databases for analyzing endgames and discovering new interesting endgames. Nunn describes different ways of using this software as follows: checking and correcting analyzed endgames; analyzing over-the-board endgames; exploring endgames to extend theory; discovering general rules that govern a certain type of ending; and forming key positions that a human player can memorize.

In this paper, we propose a model that is capable of improving some of the existing chess problems. We have formalized a major part of the knowledge needed for judging the quality of chess problems. The basic idea is to attempt to improve a given problem and its solution by a series of meaningful chess transformations using a hill-climbing search, while satisfying various criteria, step by step, until no further adequate transformations are available.

This model has been implemented in a working system called Improver of Chess Problems (ICP). The model has been tested on 36 two-move mate problems taken from chess literature. The results showed that about one-third of the problems were improved slightly. General lessons learned from this research may be useful in other composition domains.

This paper is organized as follows: the first section gives background concerning the composition of chess problems. The next section describes the proposed model, including the knowledge it uses, its subalgorithms and the various kinds of transformations it uses. The next section details three illustrative examples, each of which demonstrates other features of our system. The next section presents the results of the experiment and analyzes them and the final section summarizes the research and proposes future directions.

In the Appendix we present a detailed run of the software for the improvement process taken for the first problem discussed in the section on examples.



## COMPOSITION OF CHESS PROBLEMS

### General Background

Chess problems can be divided into two main categories:

1. Orthodox problems—problems that rely absolutely on the rules of the game. In this type of problem, White has to mate Black in a limited number of moves against any defense of Black.
2. Heterodox or fairy problems—problems that rely partially on the rules of the game. Each kind of heterodox problem has its own specific rules.

Most composers have concentrated mainly on either two or three move problems, since it has been recognized that most composition ideas can be illustrated by this kind of problem.

In this model we restrict ourselves to orthodox problems, since they are the most investigated kinds of problems and the most closely associated with the chess game. To be precise, we decided to deal with the most investigated kind of chess problems, the two-move orthodox problems.

### Two-Move Mate Problems

Two-move problems have an advantage over longer problems from the aspect of the number of variants, because they usually show more variants leading to different and interesting mate moves by White. The next section describes this kind of problem. However, in modern two-move problems, there is an increasing tendency to compose problems with complex themes, which usually limits the variety of possible solutions.

With the help of international masters in composition, we have defined a major part of the knowledge needed for evaluating the quality of chess problems in general and in two-move problems in particular (see the section on models). This knowledge includes, among other things, definitions of various composition themes in two-move problems. Most of them were collected from relevant composition books and correspondence (Haymann, 1988–1991; Howard, 1961; 1962; 1970; Harley, 1970). Ten specific themes have been implemented in our application and their definitions and relative values in points are given in Table 2. As an introduction to this table and other tables we first introduce Table 1, which explains several terms in the domain of chess problem composition.

## THE MODEL

### General Description

In this section we describe our algorithm for the improvement of mate problems. It will be presented by the following components: domain knowledge, input, improvement process, summary process, and the main flow of the algorithm.

*Domain knowledge.*

- Database of original mate problems composed by various chess composers
- Repertoire of transformations
- Heuristics responsible for applying the transformations
- Definitions of a few terms in the domain of chess problems composition (Table 1)
- Definitions of composition themes included in two-move mate problems (Table 2)
- Common relative values of chess pieces (Table 3)
- Definitions of various kinds of bonuses and penalties (Tables 4 and 5)
- Quality function for evaluating two-move mate problems (Equation 1).

*Input.*

- The number of a chosen problem from the database of problems or a given new problem.

*Improvement process.* This process is intended to attempt improving a given problem through a series of meaningful chess transformations, while satisfying several criteria at each step. The input problem is tested according to general chess and two-move mate rules. It is then analyzed automatically by a special problem analyzer in order to find its important features and its quality score.



**Table 1.** Definitions of a few technical terms in the domain of the composition of chess problems

| # of term | Term | Definition |
|---|---|---|
| 1 | Castling | A special move between a king and a specific rook of the same color. In this move the king is moved two squares towards the rook and the rook is moved over the king and placed on the adjacent square. |
| 2 | Duals (in two-move problems) | There are two possible mate moves in one variant. |
| 3 | *en passant* capture | The special capture of a pawn, which advances two squares in one move, by an opponent pawn standing near the result square. This capture is done in a diagonal movement and is allowed only immediately after the advance. |
| 4 | Key/keymove | The unique first move of White, which enables him to mate Black in the desired number of moves. |
| 5 | King-flights | Free squares that the Black king can escape to. |
| 6 | Major duals (in two-move problems) | Duals occurred in thematic variant/ions. |
| 7 | Meredith | A problem which contains at least 8 pieces and at most 12 pieces. |
| 8 | Miniature | A problem which contains at most 7 pieces. |
| 9 | Minor duals (in two-move problems) | Duals occurred in unthematic variant/ions. |
| 10 | Multiples (in two-move problems) | There are more than three possible mate moves in one variant. |
| 11 | Pinned piece | Such a piece that at least one of its moves would cause its king to be under a check (such a move will be, of course, an illegal move). |
| 12 | Thematic variant | A variant connected to the discussed theme. |
| 13 | Threat | The mate-move that White threats after making his keymove. |
| 14 | Triples (in two-move problems) | There are three possible mate moves in one variant. |
| 15 | Try | A first move of White that is answered by a single refutation of Black. |
| 16 | Unpinning | A pinned piece is released from a pin. |
| 17 | Variant | A path from the key to one of the mate positions. |

2

**Table 2.** Definitions of composition themes and their relative value in points

| # of theme | Composition theme | Definition | Value in points |
|---|---|---|---|
| 1 | Tempo or waiting move | The keymove doesn't threat any mate move | 10 |
| 2 | Direct battery | A battery with a White piece in the middle of the battery where a battery is defined as follows: a piece is standing between a long-range piece (queen, rook or bishop) and a king | 15 |
| 3 | Indirect battery | A battery to a square adjacent to the king's square | 25 |
| 4 | King-flights | The keymove creates free square/s that the Black king can escape to | 15 |
| 5 | Lonely king | The Black has only a king | 2 |
| 6 | Half-pinning | Two Black pieces standing between a White long-range piece (queen, rook or bishop) and a Black king | 25 |
| 7 | Self-pinning | The Black makes a move and pins the Black king | 15 |
| 8 | Unpin opponent piece | The Black makes a move and unpins a White piece | 20 |
| 9 | Self-blocking | A Black piece blocks another Black piece and by that creates different mate variations | 25 |
| 10 | Grimshaw | Two Black pieces which each one of them blocks the other's line and by that causes different mate variations | 45 |

This analysis is unique in the task of improving chess problems and differs in many ways from the analysis of regular chess positions, as we presented in Kerner (1995a). The analysis in our process is not only carried out for the initial problem, but also for every legal mate position during the improvement process. It includes:

- The theme/s included in the problem
- Bonuses and penalties
- The solution/s of the problem
- The problem's quality-score (Equation (1)).

Then, a specific transformation is applied to the current position at each step. The new position is examined and evaluated according to various criteria. If the new position has a higher quality score than the original problem, the new position is stored and we reapply the same process to it. Otherwise, another transformation is applied.



**Table 3.** Common relative values of chess pieces

| Piece | Queen | Rook | Bishop | Knight | Pawn |
|---|---|---|---|---|---|
| Piece value | 9 | 5 | 3 | 3 | 1 |

In order to find all basic improvements to the original problem, the application of all transformations is attempted at every step of the process. In this way, we may reach better and more complex improvements, even if a certain transformation seems not to be the best.

According to chess rules in general, and chess composition rules in particular, we must cancel all severe deficiencies. A new problem is considered an improvement on the original if it has no severe deficiencies and its quality score is higher than the quality score of the original problem. The following process achieves these improvement assignments.

After each step of the process, we apply the following checks, according to their importance in descending order:

- Severe deficiencies that cause a cancel of the position as a proposed problem:
  - Illegal chess position
  - It is not a two-move mate problem
  - More than one key/keymove
  - At least one major dual/triple
- Calculation of the problem's quality score:
  - Theme/s included
  - Number of variants
  - Number of solutions that fit the original theme/s
  - Number of mate moves obtainable from the different variants
  - Bonuses
  - Penalties
- The new problem is of a higher quality than the original problem.

The function for evaluating the quality mark of a problem is defined as follows:

$$q_m = \begin{cases} 0 & \text{Severe deficiency} \\ \sum_i V(T_i) + \sum_j V(B_j) - \sum_k V(P_k) & \text{otherwise} \end{cases} \qquad (1)$$

Table 4. Bonuses

| # of bonus | Feature | Bonus in points |
|---|---|---|
| 1 | Miniature | 10 |
| 2 | Meredith | 5 |
| 3 | Black king is in the center | 10 |
| 4 | X pieces on board | 3*(18−X) |
| 5 | Key by king | 15 |
| 6 | Mate move by king | 20 |
| 7 | Key gives X more king flights to Black | 15*X |
| 8 | Key enables Black to check White X more times | 30*X |
| 9 | Key pins a White piece | 3* piece's value |
| 10 | Key unpins a Black piece | 5* piece's value |
| 11 | Key sacrifices a White piece | 5* piece's value |
| 12 | $X_1, \ldots, X_{10}$ variation occurrences for themes #1, . . . , 10 (Table 2), respectively | First theme's occurrence gives the full value in points of the theme (as defined in Table 2). Each additional occurrence of the same theme in other variations adds only 1/5* full value. An exception is the theme "king flights." Each variation of it adds the full value of this theme. |
| 13 | Mates' value | $10^*$ (# of different mate moves)$^2$/# of variants |

Where $q_m$ is the quality measure, $V$ stands for the value function, $T_i$ the set of all themes included in the position, $B_j$ the set of all bonuses granted to the position, and $P_k$ the set of all penalties granted to the position. The various themes, bonuses, and penalties and their values are listed in Tables 2, 4, and 5, respectively. These tables have been defined with the help of international masters in composition of chess problems. Table 3 presents the common relative values of chess pieces except the king whose value is infinite (Shannon, 1950). These pieces' values are used in Tables 4 and 5.

*Summary process.* Outputs the following details to the screen and/or file:

- The given problem (initial position)



Table 5. Penalties

| # of penalty | Feature | Penalty in points |
|---|---|---|
| 1 | X minor duals | X |
| 2 | X minor triples | 2*X |
| 3 | X minor multiples | 3*X |
| 4 | Black king is in the corner | 20 |
| 5 | Black king is in the edge | 10 |
| 6 | Key is a check | 50 |
| 7 | Key is a double check | 70 |
| 8 | Key is a capture of a Black piece | 10* piece's value |
| 9 | Key is a promotion of a pawn (unless it's the theme) | 5* piece's value |
| 10 | Key takes X king flights from Black | 15*X |
| 11 | Key pins a Black piece | 5* piece's value |
| 12 | Key unpins a White piece | 3* piece's value |
| 13 | If Value of White pieces > Value of Black pieces | 2* (Value of White pieces−Value of Black pieces) |

- A summarized analysis of the given problem
- A summary of the improvement process as a sequence of transformations
- A summarized analysis for each improved problem/s (if any).

*Main flow of the algorithm.* In Figure 1 we describe the main flow of the algorithm for improving a given problem. This figure includes the following software components:

- Chess checker—checks the legality of the position according to the chess rules;
- Mate checker—tests whether a given position is a two-move mate problem. This software component uses a suitable limited search engine and checks almost all legal chess moves except two special moves—castling and *en passant* capture;
- Transformation maker—applies one of the transformations to a given position;
- Problem evaluator—analyzes the problem and computes its quality score.

4

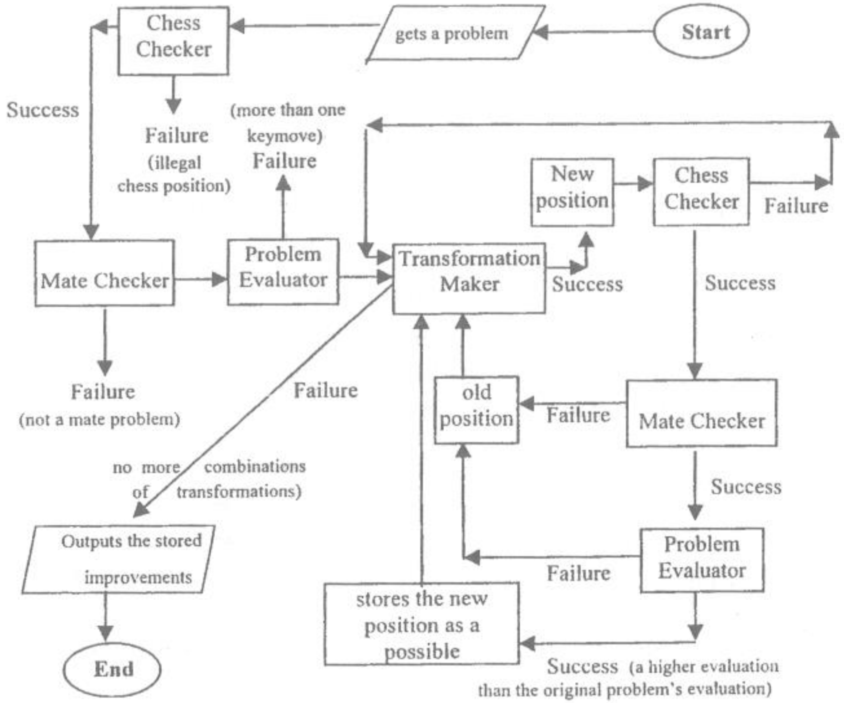



**Figure 1.** Main flow of the algorithm.

## Transformations

In this section we introduce the transformations used when attempting to improve a problem. Most of the transformations are taken from the framework introduced by Kerner (1995b).

*Simple transformations.*

- Deletion of a piece from the board—The idea behind this transformation is a known chess motive that a problem with less pieces expressing the same ideas, is of a higher quality.
- Addition of a piece to the board—The idea is that a problem with more pieces can present more ideas and, therefore, it will be with a higher quality.

*Stereotypical-agent transformations.* These strategies replace one type of piece with another piece that has the same chess impact. The




**Figure 2.** Types of stereotypical-agent transformations.

strategies are described in Fig. 2. In (a) a bishop and a queen can be exchanged because they can move along diagonals of the same color. In (b) a rook and a queen can be exchanged because they can move along either files or ranks.

Other stereotypical-agent transformations mentioned in Kerner (1995b) have not been implemented in our model, mainly because they are less important for the composition of problems. For instance, exchanging bishop and knight has not been chosen, because these pieces have different kinds of chess movements. Exchanging White and Black pieces has also not been implemented, because this replacement can be achieved by applying the two simple strategies mentioned before, i.e., deletion of a piece and addition of a piece.

*Stereotypical-area transformations.* These strategies replace one type of chess area with another type of chess area that offers the same function. The strategies are described in Fig. 3. In (a) a piece on file $i$ is transferred to file $j$ (without changing the rank). In (b) a piece on rank $i$ is transferred to rank $j$ (without changing the file).

*Transparency transformations.* These strategies transfer all pieces through a certain movement. The strategies are described in Fig. 4. In (a) all pieces are transferred $i$ files to the right side or to the left side. In (b) all pieces are transferred $i$ ranks to the upper side or to the lower side.


**Figure 3.** The stereotypical-area links.

all pieces → i files to the right / left (a)  all pieces → i ranks up / down (b)

**Figure 4.** Types of transparency transformations.

Other stereotypical-area transformations (e.g., exchanging king-side area and queen-side area) mentioned in Kerner (1995b) have not been implemented in our model, mainly because they are too general for the composition of problems.

*Application of the transformation.* To improve the given problem in an efficient way, we have constructed a priority list of transformations in descending order, responsible for the order of applying the transformations described in the previous section.

Each transformation is tried, in turn, on a specific piece on the board. Each transformation is actually applied if and only if it satisfies the following three criteria:

1. The new position is legal according to the rules of chess;
2. The new position is a two-move mate problem with only one keymove;
3. The new position has a higher quality score than the original problem.

However, we do not demand the following two criteria:

1. The new position includes the theme/s included in the original problem;
2. The new position has a higher quality score than the quality score of the best improvement found thus far.

The reason for not demanding the first criterion is because we want the model to be more general by enabling it to compose improvements not only for the theme/s included in the original problem. The reasons for not demanding the second criterion are: (1) we want to find all possible improvements, and (2) better improvements can be found based on different previous improvements.

The list below helps implement transformations in the order intended to achieve the best, most likely improvements (according to chess experts) at the earliest opportunity.



*Priority list for applying the transformations.*

1. Deletion of a piece from the board
   1.1. Deletion of a White piece from the board
   1.2. Deletion of a Black piece from the board
2. Addition of a piece to the board
   2.1. Addition of a Black piece to the board
   2.2. Addition of a White piece to the board
3. Exchanging pieces (stereotypical-agent transformations)
   3.1. Exchanging a White piece with a Black piece
   3.2. Exchanging a Black piece with a Black piece
   3.3. Exchanging a White piece with a White piece
   3.4. Exchanging a Black piece with a White piece
4. Moving a piece (stereotypical-area transformations)
   4.1. Moving the White piece, which will make the key move, as far as possible from the square it should reach by this move, by using one of the stereotypical-area transformations
   4.2. Moving other White piece/s by using one of the stereotypical-area transformations
   4.3. Moving Black piece/s by using one of the stereotypical-area transformations
5. Transparency transformations
   5.1. Moving all pieces *i* files to the right side or to the left side
   5.2. Moving all pieces *i* ranks to the upper side or to the lower side

## Complexity

Given a certain chess problem, the model is theoretically able to examine all possible legal chess positions as candidates for improvements of the given problem. All these positions can be reached, for example, by using various deletion and addition transformations. Chess problemists and mathematicians estimate the number of different legal chess positions to be $10^{40}$ (Nievergelt, 1977).

In practice, however, our model has overcome this combinatorial explosion. The number of positions the model has examined in any given problem has been reasonable to deal with. The reason for this is that we prevented the application of additional transformations when-ever a position was reached that did not satisfy the criteria mentioned in the previous section.

That is to say, our model uses a straightforward hill-climbing search. However, we are aware of the fact that this AI technique has a well-known disadvantage which is that it does not necessarily lead to a global optimum.

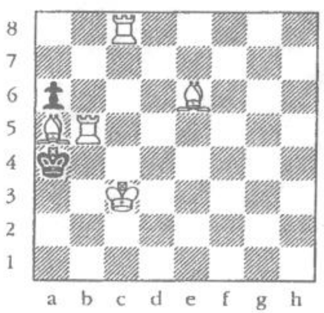
## EXAMPLES

In this section, we annotate three mate problems that have been improved by our system. The first two are taken from our database of problems. The third problem is an invented problem based on a known chess problem. Each example demonstrates other features of our system. In the Appendix we present a detailed run of the software for the first problem.

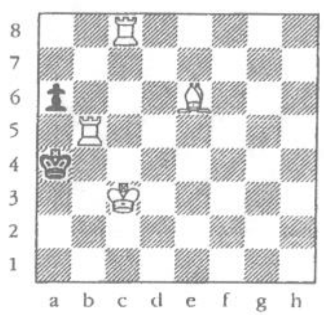
### Example 1

The first example is the miniature presented in Position 1. The composition theme expressed in this problem is "self-blocking," which means that Black himself by at least one of his moves will close his king flights enabling White to mate him.

The solution to Position 1 is as follows: The keymove is Bishop a5–d8. Then there are four variants, which can be derived from the four



Position 1



Position 2



possible answers of the Black:

| | Black's move | White's mate-move |
|---|---|---|
| a) | 1 . . . King a4–b5. | 2. Bishop e6–d7. |
| b) | 1 . . . King a4–a3. | 2. Rook b5–a5. |
| c) | 1 . . . Pawn a6–a5. | 2. Rook b5–a5. |
| d) | 1 . . . Pawn a6–b5. | 2. Rook c8–a8. |

Trying to improve Position 1, the system succeeded in operating transformation (1.1) which is "taking off a White piece from the board" by taking off the White bishop on a5. As a result, we reach Position 2. T solution to Position 2 is: the key move Bishop e6–d7. There are then three possible variants:

| | Black's move | White's mate-move |
|---|---|---|
| a) | 1 . . . King a4–a3. | 2. Rook b5–a5. |
| b) | 1 . . . Pawn a6–a5. | 2. Rook b5–b3. |
| c) | 1 . . . Pawn a6–b5. | 2. Rook c8–a8. |

The deletion of the White bishop on a5 leads to omission of the first variant for the problem in Position 1. Therefore, Position 2 has two main disadvantages in comparison to Position 1:

1. Loss of the nice mate achieved in the mentioned variant;
2. Loss of one the Black's king-flights which is 1 . . . King a4–b5 in the same variant.

However, Position 2 is considered better than Position 1 for four main reasons:

1. The same composition theme (self-blocking) is expressed by a smaller problem (with one less bishop to White, the strongest side!);
2. In contrast to the solution variants in Position 1, all solution variants in Position 2 include different mate moves. That is, all variants are rather different and each of them contributes a novelty by its mate move;
3. In addition, in Position 2 we achieved a new composition theme called "self-pinning." In the third variant the Black makes a move 1 . . . Pawn a6–b5 and pins his king;

4. Moreover, after making the key move for Position 1, White has a mate threat 2. Rook b5–a5. In Position 2 White has no mate threat, yet succeeds in mating Black in two moves. This feature is called "tempo" and it is also considered an additional composition theme.

In the Appendix we present a detailed run of the software for the improvement process taken from Position 1 (quality score = 97) to Position 2 (quality score = 118).

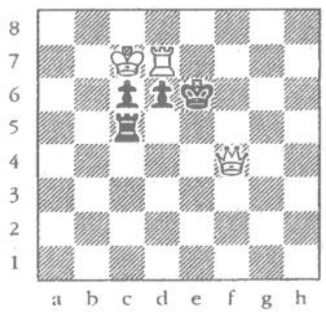
### Example 2

The second example is the miniature presented in Position 3. The composition theme expressed in this problem is "self-blocking." The solution to Position 3 is the key move: King c7–d8. Then there are 13 possible variants, only 4 of which express "self-blocking."

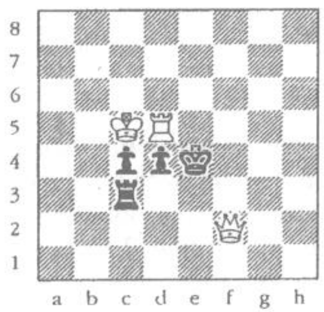
All 9 other variants include the same mate move and do not contribute any novelty. Trying to improve Position 3, the system succeeded in operating transformation (5.2) which is moving all pieces *i* ranks to the lower side. That is, all pieces on the board were moved 2 ranks below. For example, the Black king has been transferred from e6 to e4 via e5.

As a result, we arrive at Position 4. The solution to Position 4 is the key move: King c5–d6. Position 4 is considered as better than Position 3 for two reasons:

a) The Black king in Position 4 is placed on a central square which is considered harder for White to mate;



Position 3



Position 4



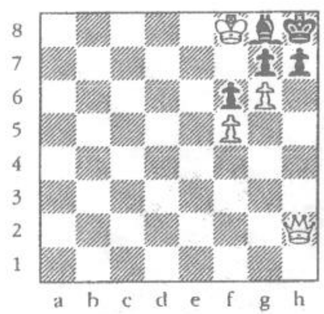
b) Position 4 includes 11 possible variants (versus 13 in Position 3), 4 of which (as in Position 3) express "self-blocking." Only 7 other variants (versus 9 in Position 3) do not contribute any novelty.

### Example 3

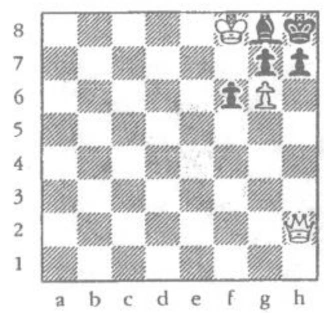
The final example is the miniature presented in Position 5. This position represents an invented problem based on a famous problem (Position 9), composed by Samuel Lloyd (1841–1911), one of the best chess composers ever. In order to test our system in applying various kinds of transformations, we took Lloyd's problem, "destroyed" it by applying some "opposite" transformations, and then attempted to improve it until we arrived at the original problem.

Starting with the problem presented in Position 5 we arrived at Lloyd's problem (Position 9) after applying four transformations as follows:

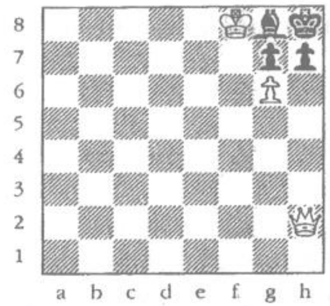
a) Deletion of White pawn on f5 (transformation (1.1)) leading to Position 6;
b) Deletion of Black pawn on f6 (transformation (1.2)) leading to Position 7;
c) Exchanging White queen to White rook (transformation (3.3)) leading to Position 8;
d) Moving the White rook from h2 to h1 (transformation (4.1)) leading to Position 9.



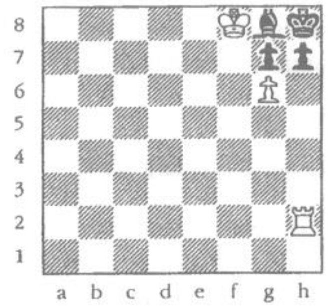
Position 5



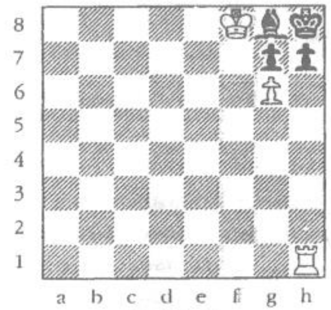
Position 6



Position 7



Position 8



Position 9

It is interesting to know that this famous Lloyd's problem (Position 9) was the first composed chess problem ever solved by a computer. It was solved by the Manchester machine in November 1951 after a few weeks of tuition. More details can be obtained in Turing et al., 1953 (pp. 295–297).

## RESULTS

We have tested our model on 36 real problems of which each included at least one theme from the 10 themes defined in Table 2. Most of the problems were collected from relevant composition books and correspondences (Haymann, 1988–1991; Howard, 1961; 1962; 1970; Harley, 1970).

The majority of the problems were miniatures; the rest were merediths. We chose to work with these kinds of problems since they include



**Table 6.** Rate of improved problems

| | Tested problems | Improved problems | Improved problems achieved after one transformation | Improved problems achieved after two transformations |
|---|---|---|---|---|
| Number of | 36 | 10 | 8 | 2 |
| Percentage | 100% | 27.7% | 22.2% | 5.6% |

a relatively small number of pieces. This feature enables us to achieve improvements to mate problems in a reasonable amount of time.

The results presented in this section relate only to the best improvements reached by the model. That is, for each problem examined, we take into consideration only the best improvement found by the model (if found).

Table 6 shows the number of problems tested, the number of problems we have improved, the number of improved problems achieved after one transformation, and the number of problems achieved after two transformations.

Table 7 presents the distribution of the transformations that have been applied to the 10 problems that have been improved. The total number of transformations in all the improved problems is 12 because in 8 of the problems one transformation was applied in each, and in 2 problems 2 transformations were applied in each one.

This experiment has shown some trends:

- The majority of the problems (26 out of 36) were optimal according to the model.

**Table 7.** Distribution of transformations leading to improvements

| | Total transformations in improved problems | Deletion | Addition | Exchange | File-transparency | Rank-transparency | Move |
|---|---|---|---|---|---|---|---|
| Number of | 12 | 4 | 0 | 0 | 4 | 3 | 1 |
| Percentage | 100% | 33.3% | 0% | 0% | 33.3% | 25% | 8.3% |

- Most of the problems improved (8 out of 10) were found by applying only one transformation. That is, these problems had been almost optimal according to the model.
- A few transformations, e.g., deletion and transparency, were successfully applied in many of the improved problems.
- A few transformations, e.g., addition and exchange, were not applied in any improvement.

## SUMMARY AND FUTURE WORK

By developing ICP, we have contributed to research in the composition of chess problems. We have developed a model that is capable of improving the quality of some of the existing chess problems from the aspect of chess composition.

The results of the experiment we have made show that about two-thirds of the examined problems were optimal, and most of the problems that were improved had been almost optimal. These results, although not apparently impressive, are relatively good considering that most of the tested problems were composed by very experienced composers.

In computer chess, high-level playing has proven inefficient without deep searching. We believe that this is also true in order to achieve a high level in composing chess problems. Therefore, adding a more complex searching technique, rather than using a hill-climbing search, would further enhance our model. That is, in order to find an improvement, we would need to allow the application of transformations even when the previously tested transformations do not satisfy the necessary criteria. In this way, after making a set of several transformations, we may reach better and more complex improvements.

Another idea is to evaluate the potential of ICP as an intelligent support system for weak and intermediate composers. Application of this idea may, in the long run reinforce ICP's strength while simultaneously improving these composers' performance.

We think that this model, in principle, can be generalized to a certain extent to all board games. The idea of applying transformations (which are in principle game-independent, e.g., deletion, addition, exchange), while satisfying specific evaluation criteria, is suitable for most



games.[1] General lessons learned from this research (e.g., formalization of knowledge, transformations, and evaluation criteria) may be useful in other composition domains.

## ACKNOWLEDGEMENTS

Thanks to Josef Retter and Jean Haymann, both international masters of the FIDE for chess composition. These two international masters supplied us with relevant background concerning composition of chess problems. Thanks also to Ephraim Nissan, Rahel Gordon, and an anonymous referee for valuable comments on this paper and to Shai Bushinski for comments on computer chess.

## REFERENCES

Harley, B. 1970. *Mate in Two Moves*. New York: Dover Publication, Inc.

Haymann, J. 1988–1991. First steps in composition. A series of correspondences. In *Variantim: Bulletin of the Israeli Chess Composition Association*.

Howard, K. S. 1961. *The Enjoyment of Chess Problems*. New York: Dover Publication, Inc.

Howard, K. S. 1962. *One Hundred Years of the American Two-Move Chess Problem*. New York: Dover Publication, Inc.

Howard, K. S. 1970. *Classic Chess Problems by Pioneer Composers*. New York: Dover Publication, Inc.

Kerner, Y. 1995a. Case-based evaluation in computer chess. In Advances in Case-Based Reasoning. *Proc. Second European Workshop*, EWCBR-94, Lecture Notes in Artificial Intelligence 984, eds. J. P. Haton, M. Keane, M. Manago, 240–254. Berlin: Springer-Verlag.

[1]The kind of processing discussed in this paper also resembles some computational approaches to the design of refueling for nuclear power reactors. A colleague in this field noted that it is customary to "shuffle" positions in a grid representing a section of the reactor; this modified configuration is then fed into a simulator (e.g., Rothleder et al., 1988). In the field of mobile radio engineering too, Kim and Chang (1994) discuss a problem of optimal allocation in a planar grid of adjacent hexagonal cells. From these concise examples, it is apparent that despite the specialized features of computer chess, such reasoning as is discussed in this paper may also be adapted to other domains.

Kerner, Y. 1995b. Learning strategies for explanation patterns. Basic game patterns with application to chess. In Case-Based Reasoning: Research and Development, *Proc. First International Conference*, ICCBR-95, Lecture Notes in Artificial Intelligence 1010, eds. M. Veloso and A. Aumodt, 491–500, Berlin: Springer-Verlag.

Kim, S., and K.-N. Chang. 1994. Optimal channel allocation for cellular mobile systems with nonuniform traffic distribution. *INFOR: Information Systems and Operational Research* 32:202–213.

Nievergelt, J. 1991. Information content of chess positions: Implications for game-specific knowledge of chess players. In *Machine Intelligence* 12, eds. J. E. Hayes, D. Michier, and E. Tyagg, 283–289, Oxford: Clarendon Press.

Nunn J. 1993. Extracting information from endgame databases. In *Advances in Computer Chess* 7, eds. H. J. van den Herik, I. S. Herschberg, and J. W. H. M. Uiterwijk, 19–34, Maastricht, The Netherlands: University of Limburg.

Rothleder, B. M., G. R. Poetschhat, W. S. Faught, and W. J. Eich. 1988. The potential for expert system support in solving the pressurized water reactor fuel shuffling problem. *Nuclear Science and Engineering* 100:440.

Shannon, C. E. 1950. Programming a computer for playing chess. *Philosophical Magazine* 41:256–277.

Turing, A. M., C. Strachey, M. A. Bates, and B. V. Bowden. 1953. Digital computers applied to games. In *Faster Than Thought*, ed. B. V. Bowden 286–310, London: Pitman.

## APPENDIX

### A Detailed Run of the Software for the Example Presented in Example 1.

THE PIECES:
white
B a 5
K c 3
R b 5
R c 8
B e 6
black
K a 4
P a 6

THE EVALUATION:
KEY: BISHOP a 5 ⇒ d 8



THREAT: ROOK b 5 ⇒ a 5

BONUSES:
problem_size = MINIATURE
pieces_number = 7
key_move_distance = 9
different_mates = 3
total_mates = 4
mates_value = 22
sacrificed_pieces = ROOK b 5

PENALTIES:
white_value = 16
black_value = 1
king_place = EDGE

THEMES:
self_blocking = 1
self_blocking variants'# = 1
variants_value = 22

POINTS:
bonuses = 129
penalties = 32
TOTAL... = 97

VARIANTS:

| BLACK ANSWER MOVE | WHITE MATE MOVE | THEME |
|---|---|---|
| 1) KING a 4 ⇒ b 5 | BISHOP e 6 ⇒ d 7 | |
| 2) KING a 4 ⇒ a 3 | ROOK b5 ⇒ a5 | |
| 3) PAWN a6 ⇒ a5 | ROOK b5 ⇒ a5 | |
| 4) PAWN a 6 ⇒ b 5 | ROOK c 8 ⇒ a 8 | self_blocking |

&&&&&&&&&& END OF EVALUATION &&&&&&&&&&

IMPROVEMENT # 1
THERE IS GOOD A TRANSFORMATION AS FOLLOWS:

11

TRANS_DEPTH: 1
TRANSFORMATION KIND: TRANS_DEL OF WHITE BISHOP a 5
TIME IS: 0.1 MINUTES

TRANSFORMATIONS MADE SO FAR:
1) TRANS_DEL OF WHITE BISHOP a 5

THE COMPARISION:

| | ORIGINAL | TRANSFORMATED | DELTA |
|---|---|---|---|
| POINTS: | 97 | 118 | 21 |
| BONUSES: | 129 | 159 | 30 |
| PENALTES: | 32 | 41 | 9 |
| PIECES_NUM: | 7 | 6 | −1 |

THE PIECES:
white
K c 3
R b 5
R c 8
B e 6
black
K a 4
P a 6

THE EVALUATION:
KEY: BISHOP e 6 ⇒ d 7

BONUSES:
problem_size = MINIATURE
pieces_number = 6
key_move_distance = 3
different_mates = 3
total_mates = 3
variants_value = 30
sacrificed_pieces = ROOK b 5



PENALTIES:
king_flights_original = 2
king_flights_after_key = 1
white_value = 13
black_value = 1
king_place = EDGE

THEMES:
tempo = 1
tempo_variants' # = 1
self_pinning = 1
self_pinning variants'# = 1
self blocking = 1
self blocking variants'# = 1
mates_value = 30

POINTS:
bonuses = 159
penalties = 41
TOTAL... = 118

VARIANTS:

| BLACK ANSWER MOVE | WHITE MATE MOVE | THEME |
|---|---|---|
| 1) KING a4 ⇒ a3 | ROOK b 5 ⇒ a 5 | |
| 2) PAWN a 6 ⇒ a | ROOK b 5 ⇒ b 3 | self_blocking |
| 3) PAWN a 6 ⇒ b 5 | ROOK c 8 ⇒ a 8 | self_pinning |

&&&&&&&&&& END OF EVALUATION &&&&&&&&&&

12