

A pattern-recognition theory of search  
in expert problem solving

Fernand Gobet  
ESRC Centre for Research in  
Development, Instruction and Training  
Department of Psychology  
University of Nottingham  
Nottingham NG7 2RD  
England

Phone: (0115) 951 5402  
Fax: (0115) 951 5324  
frg@psyc.nott.ac.uk

Gobet, F. (1997). A pattern-recognition theory of search in expert problem solving.  
*Thinking and Reasoning*, 3, 291-313.

Running head: Pattern-recognition and search

### **Abstract**

Understanding how look-ahead search and pattern recognition interact is one of the important research questions in the study of expert problem-solving. This paper examines the implications of the template theory (Gobet & Simon, 1996a), a recent theory of expert memory, on the theory of problem solving in chess. Templates are “chunks” (Chase & Simon, 1973) that have evolved into more complex data structures and that possess slots allowing values to be encoded rapidly. Templates may facilitate search in three ways: (a) by allowing information to be stored into LTM rapidly; (b) by allowing a search in the template space in addition to a search in the move space; and (c) by compensating loss in the “mind’s eye” due to interference and decay. A computer model implementing the main ideas of the theory is presented, and simulations of its search behaviour are discussed. The template theory accounts for the slight skill difference in average depth of search found in chess players, as well as for other empirical data.

## **A pattern-recognition theory of search in expert problem solving**

How much of expert problem-solving behaviour is explained by real-time search through the task problem space, and how much is explained by pattern recognition? Ever since the seminal work of De Groot (1946/1978), this question has been a recurring theme in the study of expertise (see Charness, 1991; Holding, 1985).

The study of chess players, originated in its modern form by De Groot (1946), has been a productive subfield of the psychology of expertise. Chess is a task where perception, memory, knowledge organisation and search interact in a complex, dynamic way. The advantages it offers include: a quantitative scale for measuring skill, a large database of games, and ecological validity (Neisser, 1976) of the domain. In addition, numerous empirical data have been collected, and several psychological theories of chess skill have been proposed.

The goal of this paper is to extend the template theory (Gobet & Simon, 1996a), which was originally intended as a theory of chess players' perception and memory, to the realm of problem solving. In the first part, I briefly review previous theories of problem solving in chess. Then, I propose elements of a comprehensive theory of chess problem solving that is compatible with empirical data on memory and perception research. Aspects of this theory are implemented in a computer program, SEARCH, which is described in the third part of the paper, in conjunction with results of simulations. Finally, I briefly consider how the proposed theory generalises to other domains of expertise.

### **Chess Problem Solving: Theories and Data**

#### **De Groot's classical results**

The basic finding of De Groot's study (1946/1978) was that there were no important differences in the macrostructure of search between world-class grandmasters and strong amateurs, except that the best players found better moves and were somewhat faster to reach a decision. On average, players from both skill levels searched equally deep, visited the same number of positions in their search,

and proposed the same number of candidate moves. De Groot concluded that the key to expertise is not a superiority in “speed of thinking” or in a more rational way to organise one’s thought, but specific knowledge about various aspects of the task domain. De Groot noted that the search behaviour shown by his players was compatible with Selz’s (1922) theory of productive thinking but he did not offer a detailed model of chess thinking.

### **Simon’s theory and computer programs of chess playing**

A detailed account of chess thinking was later offered by Chase and Simon (1973). Their theory, known as the chunking theory (CT), proposed that at the core of expertise lies the ability to rapidly recognise important problem features. These features, internally stored as chunks, act as access points to semantic long-term memory (LTM) and as the conditions of productions, whose actions may be carried out internally or externally. This production-system account was linked to assumptions about information-processing mechanisms, describing, for example, how an object is sorted through a discrimination net in order to reach a node in LTM. It also included parameters specifying learning and accessing times (e.g., about 8 s to learn a new chunk), as well as strict capacity limits (e.g., STM limited to 7 chunks). In the case of chess expertise, perception mechanisms in CT allow recognition of patterns of pieces on the board. These patterns suggest moves, which are used to update the internal representation of the board in what they call the *mind’s eye*. (The mind’s eye is a relational system that can be subjected to visuo-spatial mental operations and that stores perceptual structures, both from external inputs and from memory stores. ) Pattern-recognition mechanisms apply recursively in the internal representation of positions in the mind’s eye. Termination of search in a branch is obtained by evaluating whether certain goals are above or below a threshold, whose value may change as a function of players’ expectation level of the position.

Various computer programs of problem solving in chess were developed by Simon and his colleagues, but none of them directly incorporated the recognition-

association mechanism of the chunking theory. The programs written by Newell, Shaw, and Simon (1963), and Baylor and Simon (1966) implemented the idea of selective search made possible by the use of heuristics. Newell and Simon's model (1965; see also Wagner & Scurrah, 1971) formulated six principles that dictate the generation of moves or episodes, using mainly the evaluation obtained at the end of a branch.

### **Holdings' SEEK theory**

Holdings (1985, 1992) criticised CT, and proposed instead the SEEK (SEarch, Evaluation and Knowledge) theory, a theory that was however only loosely framed and did not reach the theoretical detail of CT. The main thrust of SEEK was to suggest that mastery in chess depends primarily in thinking ahead, and not in pattern recognition. Note that all three components of the SEEK theory are present in CT and that it is really the weight given to each of these components that differentiate the two theories.

### **Saariluoma's apperception and restructuring theory**

Recently, Saariluoma (1990, 1992) proposed that players, while trying to find a move in a position, access goal-positions by pattern recognition (Saariluoma uses the old term "apperception"). They then try to close the path from the problem position to the goal-position. When it is not possible to close the path, the problem space is restructured. Thus, processes leading to a Master's choosing a move may be described as a sequence of apperception-restructuring cycles, which make it possible to find solutions with only a limited search in the problem space. Saariluoma's theory, admittedly still at its early stage of development, is not quite satisfactory: it is unclear what differentiates it from Newell and Simon's (1972) means-ends analysis, and it does not make provision for what happens when players do not have a goal-position in mind.

### **Ericsson and Kintsch's long-term working memory theory**

Ericsson and Kintsch (1995) propose, within the framework of their long-term working memory theory (LTWMT), that skilled players have constructed a

hierarchical retrieval structure, not unlike the structure created by SF and DD, two mnemonists specialised in the digit-span task (Chase & Ericsson, 1982). This powerful memory structure, isomorphic to the 64 squares of the chess board, allows players to update information rapidly during search. The theory also proposes that chessplayers can create new LTM associations rapidly, although no parameter is given to specify this speed of encoding.

I have shown elsewhere (Gobet, 1996) that LTWMT, in addition to being vague and leaving many details unspecified, generates predictions that do not agree with the experimental data on chess perception and memory. In addition, the memory mechanisms proposed by LTWMT (retrieval structure and rapid elaboration of LTM structures) seem at odds with the following aspect of chess players' search behaviour. It is common for chess players to revisit a node during their analysis of the position. However, instead of accessing this node directly, as suggested by LTWMT's very powerful mechanisms for storing positions rapidly and accurately, players access the target position by generating a sequence of individual moves from the given problem position (cf. De Groot, 1946). This behaviour suggests that players may not store positions as fast as proposed by LTWMT.

### **The Template Theory and its Application to Problem Solving**

Initially, the template theory (TT) was developed to remove some weaknesses in the chunking theory with respect to data from memory research (Gobet & Simon, 1996a). Contrary to CT's predictions, based on a limited STM capacity, chess players' memory is resistant to retroactive interference (Charness, 1976) and chess players can recall several briefly-presented positions with reasonable accuracy (Cooke, Atlas, Lane & Berger, 1993; Gobet & Simon, 1996a).

Like CT, TT proposes that chunks are accessed by traversing a discrimination net. Chunks are linked to other information stored in LTM, such as moves, plans, tactical motives, and so on. In addition, chunks that often occur in a player's experience can evolve into more complex data structures (templates), which have *slots* allowing variables to be encoded, for example, to specify the square occupied

by a piece. According to TT, this encoding occurs rapidly (about 1 s per slot). In addition to slots, templates contain a *core*, which is made up of (variable-free) chunks. Altogether, templates are assumed to store at least about ten pieces. Finally, templates may be linked to other templates. TT also proposes that information stored in the mind's eye, a structure similar to that proposed by CT, decays rapidly, and that it needs to be updated either by inputs from the external world or by inputs from memory structures.

TT removes the difficulty of CT with interference phenomena, first because it hypothesises some fast LTM encoding (through template slots), and second because templates contain more information than the chunks proposed by CT (CT assumed that chunks contain at most five or six pieces). Finally, TT keeps the power of CT to explain the rapid recognition of positions shown by masters, since the same discrimination mechanisms are present (Gobet, 1996).

### **Search mechanisms**

TT and CT propose that search is carried out in a forward fashion, by recursive application of pattern recognition processes in the internal representation. In the remainder of this section, I will flesh out this mechanism with additional specifications. I will also show how chunks and templates may be used to terminate search, and how templates may be used to carry out search at a level more abstract than the move level.

#### **Move generation.**

As mentioned above, pattern recognition is applied recursively in TT. If a chunk—or a template, since templates are a special case of chunks—is recognised, it may give access to a move (or a sequence of moves) or to an heuristic for selecting the next move. Examples of such heuristics are: "Occupy open lines," or "Counter-attack in the centre when attacked on the King's side." In this case, the move(s) is (are) carried out in the mind's eye. If no move results from pattern recognition, one of two operations may be applied. Either there is a new attempt to generate a move by accessing a different chunk, or weaker heuristics are applied.

### **Update of internal representation and storage of information into template.**

When a move or a sequence of moves is carried out in the mind's eye, the internal representation is updated, and, when a template has been accessed, slots are updated as well. Slots may store information about the location of pieces, strategic information, evaluation of the position or part of it, and so on. Note that this information facilitates search, because possible forgetting through decay in the mind's eye may be backed-up by information stored in slots.

### **Evaluation of nodes.**

Evaluation of a position is not carried out after each move but only at the terminal positions. Evaluation is either retrieved, when possible, from chunks that have been recognised, or computed when the player decides to terminate search (see next paragraph). De Groot's (1946) protocols indicate that players evaluate only terminal positions and that their evaluations are rather simple, taking into account a single feature, such as mobility or control of the centre, and strongly suggest that evaluations are not computed by conjoining several features.

### **Ending search of an episode.**

A critical question, given a limited information-processing capacity, is when to stop searching an *episode* (i. e., a sequence of moves generated from a base move). I propose that the following conditions can halt the search of an episode. To begin with, the probability of making an error is too high. Psychologically, this probability is a function of the forgetting and interference rates in the mind's eye. Phenomenologically, players probably assess this probability by taking into account the depth of search reached and by estimating their familiarity with the position.

Another reason to halt search occurs when the level of expectation for the position is met or when a feature of the position (e.g., safety of the king) was evaluated to be lower than its admissible value (Newell & Simon, 1972). This level of expectation is provided by a chunk encountered during earlier search or by explicit evaluation of the position, and was stored either in STM or, when available,

in a template slot. A final halting condition is that a decisive event occurred such as mate, important material gain or loss, or fulfilment of a key goal.

#### **Selecting the next base-move.**

Now consider the question of which base-move to search next when a leaf node has been reached. Any of six mechanisms may be engaged. First, if the leaf was evaluated with a high risk of error, the same sequence of moves is searched again in order to obtain a more reliable estimate. Second, the same sequence of moves may be searched again to make sure that it has been stored in LTM. Third, if a template has been recognised, a meta-search (see below) is carried out. Fourth, there is a bias to give preference to normal base-moves early in search, and to unusual base-moves later (Newell & Simon, 1972, p. 723). Fifth, a move reached at any level of depth in searching a line may be tried as a candidate move in the next line. Sixth, if the evaluation of an episode gives a favourable result, its analysis is continued; if the evaluation is unfavourable a different base-move is taken up (Newell & Simon, 1972).

#### **Meta-search with templates (abstract planning).**

Search may be carried out not only at the move level, but also at the template level. Players sometimes develop a tree search by generating key positions (templates) instead of developing a tree search by generating moves (e.g., De Groot, 1978, p. 153). Paths between key-positions consist of macro-operators or in unspecified operators, which will be filled in at a later stage of search, if the leaf positions are judged favourable. This type of search is contingent to the presence, in players' LTM, of templates sufficiently close to the position at hand. Koedinger and Anderson (1990) have proposed a similar idea with respect to geometry. This meta-search mechanism is also similar to the "apperception" mechanism suggested by Saariluoma (1990, 1992).

#### **Theoretical importance of templates**

Obviously, many of the processes described above can be implemented within the earlier CT without hypothesising the presence of templates. Templates are,

however, important for some of the key cognitive processes in chess, because (a) slots allow information about strategic or tactical themes, as well as about evaluation of the position, to be encoded in LTM rapidly; (b) links between templates allow for “meta-search;” (c) information stored both in the core and in the slots of the template compensates for decay and interference in the mind’s eye, as it may be used to reinstate fading information. This compensation mechanism is particularly useful when perceptual information from the external board is unreliable, for example because several moves have been carried out in the mind’s eye.

### **Empirical data supporting the TT account of search and pattern recognition**

In this section, TT is qualitatively applied to some of the key data from empirical studies of chess problem-solving. Quantitative assessment will be the topic of the following section. I have shown elsewhere (Gobet, 1996b) that the predictions of TT are well supported by empirical data from research on memory and perception.

Increasing the number of chunks and templates has two opposite effects with respect to search (Gobet, in press). On the one hand, deep search is facilitated, as compared with the more computationally expensive weak methods, because pattern recognition is a rapid process and because sequences of moves may be learnt and suggested by chunks. On the other hand, the need for deep search often disappears because evaluations suggested by pattern recognition may be more reliable than evaluations computed by weak methods. These two opposite effects can explain why the skill differences in average depth of search are small.

Pattern recognition allows strong players to be highly selective in their search. This can be seen both in the rapid way masters zero in into the key aspects of a position (De Groot, 1946), and in the way eye movements differ as a function of skill during the first seconds of the perception of a position (De Groot & Gobet, 1996). Additional evidence is provided by Saariluoma (1990), who has shown that players choose stereotyped solutions in problems where faster (in the number of moves to reach mate) but uncommon solutions are present. These data add support to TT, because pattern recognition lies at its core.

Further strong evidence for the role of pattern recognition is that Masters can play several opponents simultaneously at a strength close to that in one-against-one play. It is interesting that, in simultaneous games, they tend to play “normal” moves and rely on opponents’ errors (Gobet & Simon, 1996c). Normal moves may correspond to the type of moves suggested by chunks and templates.

A feature of chess expertise captured uniquely by TT is the high-levels at which chess players organise their knowledge. De Groot (1946) proposed that players organise positions around large “complexes,” bigger than the chunks that were later proposed by Chase and Simon (1973). De Groot also suggested, as did Cooke et al. (1994), that players represent information at a conceptual level. In particular, De Groot (1946) proposed that Masters’ descriptions of games are centred around key positions of the game. TT offers theoretical structures (templates) that could account for players’ large complexes, high-level representations, and key positions. TT also offers mechanisms, based on frequency of occurrences, explaining how these high-level representations are developed, and mechanisms, based on discrimination processes, explaining how they could be accessed rapidly.

TT is also compatible with results from experiments using positions for which access to knowledge is difficult. In Holding and Reynolds’ (1982) experiment, chess skill did not correlate with the recall of semi-random positions shown for a few seconds, but effectiveness of the search for the best move in these positions did correlate with skill. TT offers an explanation for the fact that there was a strong correlation between skill level and quality of move proposed. Players recursively generate a move through weak methods when no chunk is recognised, until one of the halting conditions (mentioned earlier) applies. Then, given their larger database of chunks, strong players are more likely than weak players to find a useful pattern in positions that have been newly updated in the mind’s eye, and therefore are more likely to find good moves in their search.

I mentioned above that Ericsson and Kintsch (1995) proposed mechanisms (retrieval structure and elaborations of LTM schemas) that encode information fast

and reliably. By contrast, the encoding capacity of templates is relatively weak. In particular, TT predicts that the same template may sometimes be used to encode several similar positions, thus increasing the risk of interference as search progresses. As a consequence, and contrary to LTWMT's predictions, players should often revisit positions by generating moves instead of accessing it directly from memory. This is what players – even masters – do (De Groot, 1946). Finally, TT may have something to say about errors in chess. Krogus (1976) mentions that errors often occur when a move has caused important changes in the position. This may be due to players' inability to recognise that a previously adequate template has become inadequate.

In summary, TT offers a good qualitative account of several important chess phenomena. It is now time to turn to more objective, quantitative tests of the theory.

### **A Computer Model of Search in Chess**

In the remainder of this paper, I present a computer model called SEARCH, which implements TT in some detail. I first describe the implementation of SEARCH, making clear what aspects of TT have been included, and then discuss its behaviour with respect to human data.

#### **Overview of model**

Full implementation of TT as a chess-playing computer program would be a major task – this difficulty is demonstrated by the fact that no complete computer model of chess cognition, including perception, memory and problem solving, has ever been developed. I pursued a limited aim: a computer model that contains some of the important parameters of TT and that computes key behavioural variables. This model does not constitute a complete implementation of the theory (it does not search or evaluate chess positions), but is detailed enough to offer a good test of some aspects of TT. It is sobering to notice that even this partial implementation of the theory is quite complex and requires many parameters. This is the price of avoiding the vagueness of verbal theorising.

#### **The main problem-solving loop**

The main interest of the model is in the interaction between pattern recognition and search. The model generates independent episodes following several rules for generating moves and choosing stop conditions. Compatible with most human protocols, the program considers only one branch within an episode (thus, there is no branching point within an episode). As illustrated in Figure 1, the model is close to the description of TT given earlier in this paper. The various conditions and operations in SEARCH are represented as probabilities or probability distributions, and not as actual chess moves, evaluations, or symbolic structures in STM or in the mind's eye.

A trace of SEARCH is shown in Figure 2. The analysis will centre on the following variables as a function of the number of chunks: rate of generating a move; level of fuzziness in the mind's eye when an episode is ended; mean depth of search; methods used for generating a move; and methods used for stopping an episode.

-----  
Insert Figure 1 about here

-----  
-----  
Insert Figure 2 about here  
-----

The concept of template is implemented in two ways: (a) by higher probabilities of finding (sequences of) moves and evaluations than with plain chunks, and (b) by a decrease in the level of fuzziness, made possible by the LTM representation of the core of the position and of slots.

The level of reliability of the information in the mind's eye is captured by a "fuzziness" parameter, expressed in arbitrary units. Three main cognitive features are captured by this parameter: working memory capacity, ability to maintain a visuo-spatial image, and subjective estimate of the risk of making an error. The fuzziness parameter is then assumed to originate both from the hardware and the

software of the system. It may vary during the lifetime of a player, because of biological maturation or ageing, and may be affected by experimental manipulations. There are three reasons why SEARCH ends a search: the level of fuzziness is too high, an evaluation is proposed, or no move is proposed. The idea of expectation level is not explicitly present in the model, but it is implicit in the probability values of finding features ending the search of an episode, either by pattern recognition or by the application of heuristics. Selection of the next base-move (a key concern of Newell & Simon, 1965, and Wagner & Scurrah, 1971) is not addressed by SEARCH, which in general lacks the semantic information necessary for such a selection. Finally, meta-search with templates is not directly implemented in SEARCH, but it is implicitly represented by the relatively high probability of finding (sequences of) moves and evaluations with templates, as compared with chunks.

### **The model's parameters**

SEARCH has many parameters and it is, of course, necessary to set up a few of them with independent data in order to reduce the number of degrees of freedom in the model. When possible I have used simulations with the CHREST (Gobet, 1993; De Groot & Gobet, 1996) and CHUMP (Gobet & Jansen, 1994) programs to set parameter values. CHREST is a program implementing aspects of TT and simulating numerous experiments on perception and memory in chess, and CHUMP is a program playing (weak) chess by pure pattern recognition.

#### **Number of chunks and templates.**

Table 1 shows the values that will be used with the SEARCH model for these parameters. To choose the values related to the number of chunks and templates, the current version of CHREST has been employed. The probability of finding a template was estimated by running CHREST on 50 positions randomly taken from masters' games. The first line (0 chunk) was added to get predictions at the novice level. With nets of 100,000, 120,000 and 150,000 chunks, the parameters were estimated with the best-fitting functions available (the "novice" values were not

used). The number of templates is roughly a linear function of the number of chunks ( $r^2 = 0.97$ ). The probability of finding a template is approximately a log-function of the number of chunks ( $r^2 = 0.94$ ).

-----  
Insert Table 1 about here  
-----

**Heuristics, chunks, templates, and their relation to the probability of finding moves and evaluations.**

Table 2 gives the values used in the simulations for the parameters related to heuristics, chunks, and templates. Heuristics include simple search techniques (e.g., “take a piece back” or “check the King”) as well as more abstract rules (e.g., “occupy the centre”). Their main characteristics, in comparison with chunks and templates, is that they are general and require some conscious computation in order to produce a move. The assumption underlying Table 2 is that the parameters related to heuristics stay roughly constant, while the other parameters vary as a logarithmic function of the number of chunks in the net. (A logarithmic function has been chosen to indicate that chunks yield a diminishing return as their number increases.) The assumption that the number of heuristics is constant across skill levels seems reasonable (perhaps with the exception of the novice level), as most chess masters’ knowledge is very specific to certain aspects of the game (De Groot, 1946).

The following considerations were made in setting the values given in Table 2. It was deemed plausible that the probability of finding some heuristic in a position be high (.95), as well as the probability of finding some chunk. Finally, the probability of recognising a template in a position is smaller and is given in Table 1.

The probability of finding a move given a heuristic is set to .50, indicating that heuristics may fail to produce a move half of the time. This is because some heuristic may not yield a move and because some proposed moves may be illegal. The probability of finding a move given a template is high, partly reflecting the rich

semantic knowledge and the inter-template links assumed to characterise templates. For the probability of finding a move, given that a chunk has been recognised, I used the parameter *prob* in the function *g* (see Table 2) so that it approximates the data produced by the CHUMP program (Gobet & Jansen, 1994). Evaluations are supposed to be harder to learn and, therefore, harder to use than moves. Hence, their probabilities are lower than the corresponding probabilities with moves. Again, following directly from the theory, templates offer the best source of evaluations.

-----  
Insert Table 2 about here  
-----

With respect to the number of sequential moves proposed, counted in *plies* (i. e., moves for White *or* Black), it is assumed that heuristics give access to only one move, while chunks and templates yield sequences of moves whose number follows a geometric probability distribution (the choice of this distribution was inspired by statistics taken from CHUMP). The longest sequence (call it *k*) has a frequency of 1, the second-longest a frequency of 2, the third-longest a frequency of 4, and so on until the shortest sequence, whose length is 1 move and whose frequency is  $2^{(k-1)}$ . The maximum sequence of moves was set to 18. Finally, SEARCH attempts to generate a move by pattern recognition (chunk or template) four times before using a heuristic.

#### **Time and fuzziness parameters.**

The time parameters for the respective cognitive operations were set using plausible values (see Table 3). Move generation by pattern recognition (chunk or template) is assumed to be very short (a few hundred milliseconds); no time cost is used in the model. The time for computing a move-generating heuristic was set to the rather long 20 s – generating a move without using pattern recognition requires putting together various types of information and ask for many checking procedures. The time for carrying out one move (once generated) in the mind's eye

was set to 2 s. This is compatible with Saariluoma's (1989) data, which show that a blindfolded master can follow a game dictated at 2 s a move. Finally, the time for carrying out an evaluation by heuristic is set to 10 s. This rather short time may be justified by the fact that players typically consider only one (subjective) aspect of the position (De Groot, 1949) and that many evaluations can be based on the basis of clear material losses.

-----  
Insert Table 3 about here  
-----

The fuzziness level of the system is expressed in arbitrary units. It starts with the value zero, and it is increased when a move is carried out in the mind's eye and when time is spent applying a heuristic. When a template is found, the fuzziness level is decreased, since part of the information may be stored in LTM directly, freeing capacity in working memory and in the mind's eye.

### **Simulations with the "canonical model"**

The simulations described in this section are straightforward. Using the parameter values discussed above, I let the program search 1,000 episodes for each net size. Summary values for mean depth of search, time, ratio depth/time, and fuzziness, as well as for generation and stop conditions were computed for each net size.

#### **Mean depth of search, rate of search, and fuzziness.**

-----  
Insert Figure 3 about here  
-----

According to SEARCH, mean depth of search follows a power function of the number of chunks (see Figure 3a). If we assume a constant rate of learning new chunks, as is common practice in some cognitive architectures (cf. Newell, 1990), and if we assume that skilled players have spent more practice and study time than unskilled players, then mean depth follows a power function of skill as well. This

prediction, which is at variance with Holding's prediction (1985, p. 182) of a linear increase, is partly consistent with Charness' (1981) proposal that depth of search is not linearly related to skill, but that there is a ceiling at high skill levels. The result is also compatible with the weak linear relation found in empirical studies (Charness, 1981; Gobet, in press) between mean depth of search and skill level, because restricted ranges of the curve can be approximated by a linear function. The absolute values are also quite reasonable: they rise from low depths (less than 1 ply) with no chunks (corresponding to the novice level) to around 2.5 plies with a small net (1000 chunks), and then to around 4 plies with expert level (around 30,000 chunks, as estimated by simulations on the recall task). They then slowly (save for a few random variations) increase to more than 5 plies with net larger than 100,000 chunks, which corresponds to Grandmaster level. Holding (1985) computed mean depth of search for de Groot's (1946) data and found that Experts and Grandmasters had respectively an average depth of search of 4.8 and 5.3. The regression equation derived from Charness' (1981) data yield an average depth of search of about 2.3 plies at the 1300 ELO<sup>1</sup> level, of 4.1 at the expert level (2000 ELO), and of 5.7 at the Grandmaster level. Unfortunately, no study tested players ranging from novices to Grandmasters, and SEARCH's prediction of a power law of practice cannot be tested (yet). I will report below simulations addressing data collected by Saariluoma (1990), where International Masters and Grandmasters tended to search less than Masters.

The rate of search also follows a power function of the number of chunks (see Figure 3b). De Groot (1946), Charness (1981a) and Gobet (in press) all found that strong players tended to search faster. For example, Gobet (in press) found that his Masters, Experts, Class A and Class B players respectively searched 4.8, 4.1, 3.2 and 3.4 nodes per minute. SEARCH's values are higher, but still plausible. For example, Wagner and Scurrah 's (1971) Expert generated about 10 moves per minute. Note also that all the experiments mentioned above have used verbal protocols, which may have slowed down the rate of search (cf. Ericsson & Simon, 1980). Finally, the

rate of search may have been underestimated in the literature, because the rate of generating nodes per minute is typically computed by dividing the number of nodes by the total time. However, total time includes time not spent in search (for example, what De Groot calls the “first phase”).

The plausibility of the simulations of the last variable, the level of fuzziness at which a search is ended, is more difficult to judge, because little is known about the role of fuzziness in chess thinking. Again, SEARCH proposes that this variable is a power function of the number of chunks (see Figure 3c). With larger nets, episodes tend to be ended with lower levels of fuzziness. This suggests that, in general, strong players should be more confident in their calculations, as they are not at the edge of their possibilities. Note that the rate of change is rather high for this power law, which suggests that it should be possible to detect it experimentally.

#### **Generation of moves.**

How is move generation affected by skill level? As shown in Figure 4a, SEARCH makes clear predictions. With small numbers of chunks, the time-consuming process of generating moves by heuristics is employed. This process is soon dominated by generating moves by chunks, which is in turn dominated by template generation.

-----  
Insert Figure 4 about here  
-----

#### **Stopping an episode.**

Again, SEARCH makes clear predictions about the way episodes are stopped (see Figure 4b). At first, about 65% episodes are stopped because no move is proposed and a little less than 25% are stopped because the information in the mind’s eye and in working memory is too fuzzy. As the number of chunks increases, the proportion of episodes ended by evaluation (through chunk or template recognition, as well as through heuristics) increases to about 75% with large nets.

### Variability of depth of search

In Saariluoma's (1990) experiment, International Masters and Grandmasters tended to search less than relatively weak Masters (around 2200 ELO). When plotted against skill, both the total number of nodes searched and the mean depth of search showed an inverted U-curve, with Masters searching the largest number of nodes (52) and at the largest average depth (5.1 moves). By comparison, Saariluoma's top-level players searched through a space of 23 nodes with an average depth of 3.6 moves.

Note that, as is unfortunately the rule in chess research, Saariluoma used few players and few test positions. Thus, one can expect large fluctuations just by chance. This is actually how SEARCH behaves when the number of trials is low. Figure 5 illustrates the results of simulations with the number of episodes for each net size limited to 50, which roughly corresponds to the total produced by five players searching an average of 10 episodes (Saariluoma imposed a ten-minute limit to his players). The other parameters are the same as in the canonical model.

-----  
Insert Figure 5 about here  
-----

While the curves can be well fitted by a power function, one can observe important variations from one curve to another. All curves show an increase, with dips and peaks, and all predict that some category of stronger players will search less deeply than some category of weaker players. The best example with respect to Saariluoma's data is the bottom curve. His (weak) Masters may be compared to the 40,000-chunk net, which shows deeper search than the larger nets, including the nets corresponding to grandmasters.

Thus, Saariluoma's data are consistent with SEARCH and its prediction that depth of search is a power function of skill. That Masters search deeper than top-level players may be accounted for by random fluctuations, as shown in the simulations.

### **SEARCH: Summary and conclusions**

A computer model of search in chess has been presented, based on the template theory. The model employs heuristics and pattern recognition (both of chunks and templates) to generate independent episodes. While it is assumed that the parameters related to heuristics are constant across skill levels, it is proposed that the parameters related to pattern recognition vary as a logarithm function of the number of chunks. Other parameters related to time needed to carry out an operation and to level of fuzziness in the system are also used. The model accounts reasonably well for several empirical data, both qualitatively and quantitatively (e.g., mean depth of search and rate of search). It also sheds light on Saariluoma's (1990) empirical results, which previously seemed anomalous.

Perhaps more importantly, the model provides some new theoretical insight with respect to the role of chunks in problem solving. Chase and Simon's chunking theory, although simple in appearance, has unexpected consequences due to the non-linear dynamics involved in the acquisition of chunks. For example, it had been overlooked for 25 years that CT (correctly) predicts a skill effect in the recall of random positions (Gobet & Simon, 1996b). The current paper shows that, contrary to a common interpretation of CT, acquisition of more chunks leads to deeper search.

The model also draws attention to search features not heeded in current research and poses new empirical questions. For example, what is the role of the fuzziness parameter, and what is its relation to working memory? Are variables such as depth and rate of search best approximated by a power function of skill, ubiquitous in cognitive psychology, or by a linear function, as proposed, among others, by Charness (1981) and Holding (1985)? Finally, the model is a first attempt to rigorously link a model of chess problem solving to a theory of memory and perception (the template theory and its CHREST implementation), thus providing a first step into a unified, computational theory of chess players' cognition.

The SEARCH model admittedly tackles only a subset of problem solving – it may be complemented by Newell and Simon’s (1965) or Wagner and Scurrah’s (1971) models – but makes precise and clear predictions. This is a distinct advantage in comparison with other current theories of chess problem solving, such as SEEK (Holding, 1985), which are typically vague.

To what extent do the theoretical ideas put forward in this paper generalise to other domains of expertise? One of the main teachings of recent research is that different task environments tax cognitive functions differently. In particular, environments differ massively in the amount of search needed to carry out the task successfully. Hence, it is unlikely that findings from chess research will apply without qualification to other domains. However, the interactions between low-level and high-level representation, on the one hand, and between knowledge and search, on the other hand, are also apparent in several domains, such as medical diagnosis, physics, and text comprehension. The template theory, which accounts for these interactions, thus offers the perspective to integrate these various domains of expertise into a single framework.

The chunking theory has sometimes been misinterpreted as a claim that recognition of familiar patterns and retrieval of moves associated with them is almost the sole basis of expertise in chess (e.g., Holding, 1985). The correct claim, and the one actually made by Chase and Simon, is that skill in playing chess depends both on (a) recognising familiar chunks in chess positions when playing games, and (b) exploring possible moves and evaluating their consequences. Hence, expertise depends both on the availability in memory of information about a large number of frequently recurring patterns of pieces, and upon the availability of strategies for highly selective search in the move tree. Expert memory, in turn includes slowly acquired structures in long-term memory (templates) that augment short-term memory with slots (variable places) that can be filled rapidly with information about the current position.

### Reference list

- Baylor, G. W., & Simon, H. A. (1966). A chess mating combinations program. *Proceedings of the 1966 Spring Joint Computer Conference*, Boston, 28, 431-447.
- Charness, N. (1976). Memory for chess positions: Resistance to interference. *Journal of Experimental Psychology: Human Learning and Memory*, 2, 641-653.
- Charness, N. (1981). Search in chess: Age and skill differences. *Journal of Experimental Psychology: Human Perception and Performance*, 2, 467-476.
- Charness, N. (1991). Expertise in chess: The balance between knowledge and search. In K. A. Ericsson & J. Smith (Eds.), *Studies of expertise: Prospects and limits*. Cambridge: Cambridge University Press.
- Chase, W. G., & Ericsson, K. A. (1982). Skill and working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 16). New York: Academic Press.
- Chase, W. G., & Simon, H. A. (1973). The mind's eye in chess. In W. G. Chase (Ed.), *Visual information processing*. New York: Academic Press.
- Cooke, N. J., Atlas, R. S., Lane, D. M., & Berger, R. C. (1993). Role of high-level knowledge in memory for chess positions. *American Journal of Psychology*, 106, 321-351.
- De Groot, A. D. (1946). *Het denken van den schaker*. Amsterdam: Noord Hollandsche.
- De Groot, A. D. (1978). *Thought and choice in chess*. (Revised translation of De Groot, 1946; 2nd ed.). The Hague: Mouton Publishers.
- De Groot, A. D., & Gobet, F. (1996). *Perception and memory in chess: Heuristics of the professional eye*. Assen: Van Gorcum.
- Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, 102, 211-245.
- Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87, 215-251.
- Gobet, F. (1993). *Les mémoires d'un joueur d'échecs* [The memories of a chess player]. Fribourg (Switzerland): Editions Universitaires.
- Gobet, F. (1996). *Memory in chess players: Comparison of four theories*. (Tech. Rep. No. 43). University of Nottingham (UK): Department of Psychology, ESRC Centre for Research in Development, Instruction and Training.
- Gobet, F. (in press). Chess thinking revisited. *Swiss Journal of Psychology*.
- Gobet, F., & Jansen, P. (1994). Towards a chess program based on a model of human memory. In H. J. van den Herik, I. S. Herschberg, & J. W. Uiterwijk (Eds.), *Advances in Computer Chess 7*. Maastricht: University of Limburg Press.
- Gobet, F., & Simon, H. A. (1996a). Templates in chess memory: A mechanism for recalling several boards. *Cognitive Psychology*, 31, 1-40.

- Gobet, F., & Simon, H. A. (1996b). Recall of rapidly presented random chess positions is a function of skill. *Psychonomic Bulletin & Review*, 3, 159-163.
- Gobet, F., & Simon, H. A. (1996c). The roles of recognition processes and look-ahead search in time-constrained expert problem solving: Evidence from grandmaster level chess. *Psychological Science*, 7, 52-55.
- Holding, D. H. (1985). *The psychology of chess skill*. Hillsdale, NJ: Erlbaum.
- Holding, D. H. (1992). Theories of chess skill. *Psychological Research*, 54, 10-16.
- Holding, D. H., & Reynolds, R. I. (1982). Recall or evaluation of chess positions as determinants of chess skill. *Memory & Cognition*, 10, 237-242.
- Koedinger, K. R., & Anderson, J. R. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, 14, 511-550.
- Krogius, N. (1976). *Psychology in chess*. London: R.H.M Press.
- Neisser, U. (1976). *Cognition and reality. Principles and implications of cognitive psychology*. San Francisco: Freeman & Company.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Newell, A., Shaw, J. C., & Simon, H. A. (1963). Chess-playing programs and the problem of complexity. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York, McGraw-Hill.
- Newell, A., & Simon, H. A. (1965). An example of human chess play in the light of chess-playing programs. In N. Weiner & J. P. Schade (Eds.), *Progress in biocybernetics*. Amsterdam: Elsevier.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Saariluoma, P. (1989). Chess players' recall of auditorily presented chess positions. *European Journal of Cognitive Psychology*, 1, 309-320.
- Saariluoma, P. (1990). Apperception and restructuring in chess players' problem solving. In K. J. Gilhooly, M. T. G. Keane, R. H. Logie & G. Erdos (Eds.), *Lines of thinking* (Vol. 2). New York: Wiley.
- Saariluoma, P. (1992). Error in chess: The apperception-restructuring view. *Psychological Research*, 54, 17-26.
- Selz, O. (1922). *Zur Psychologie des produktiven Denkens und des Irrtums*. Bonn: Friedrich Cohen.
- Wagner, D. A. & Scurrah, M. J. (1971). Some characteristics of human problem-solving in chess. *Cognitive Psychology*, 2, 454-478.

## Author Notes

I am grateful to Frank Ritter and to an anonymous reviewer for helpful comments on this paper. Correspondence concerning this article should be addressed to Fernand Gobet, Department of Psychology, University of Nottingham, Nottingham NG7 2RD, England.

Figure 1. Flowchart of SEARCH.

Figure 2. Examples of episodes generated by SEARCH.

Figure 3a. Mean depth of search (in plies) as a function of the number of chunks in the discrimination net.

Figure 3b. Rate of search (in move per minute) as a function of the number of chunks in the discrimination net.

Figure 3c Level of fuzziness (in arbitrary units) with which episodes are ended as a function of the number of chunks in the discrimination net.

Figure 4a. Area graph of the proportion of moves generated by heuristics, chunks or templates, as a function of the number of chunks in the discrimination net.

Figure 4b. Area graph of the proportion of episodes stopped because the level of fuzziness was too high, because the position was evaluated, or because no move was proposed, as a function of the number of chunks in the discrimination net.

Figure 5. Mean depth of search (in plies) as a function of the number of chunks in the discrimination net, in five simulations where the number of episodes is small ( $n=50$ ).

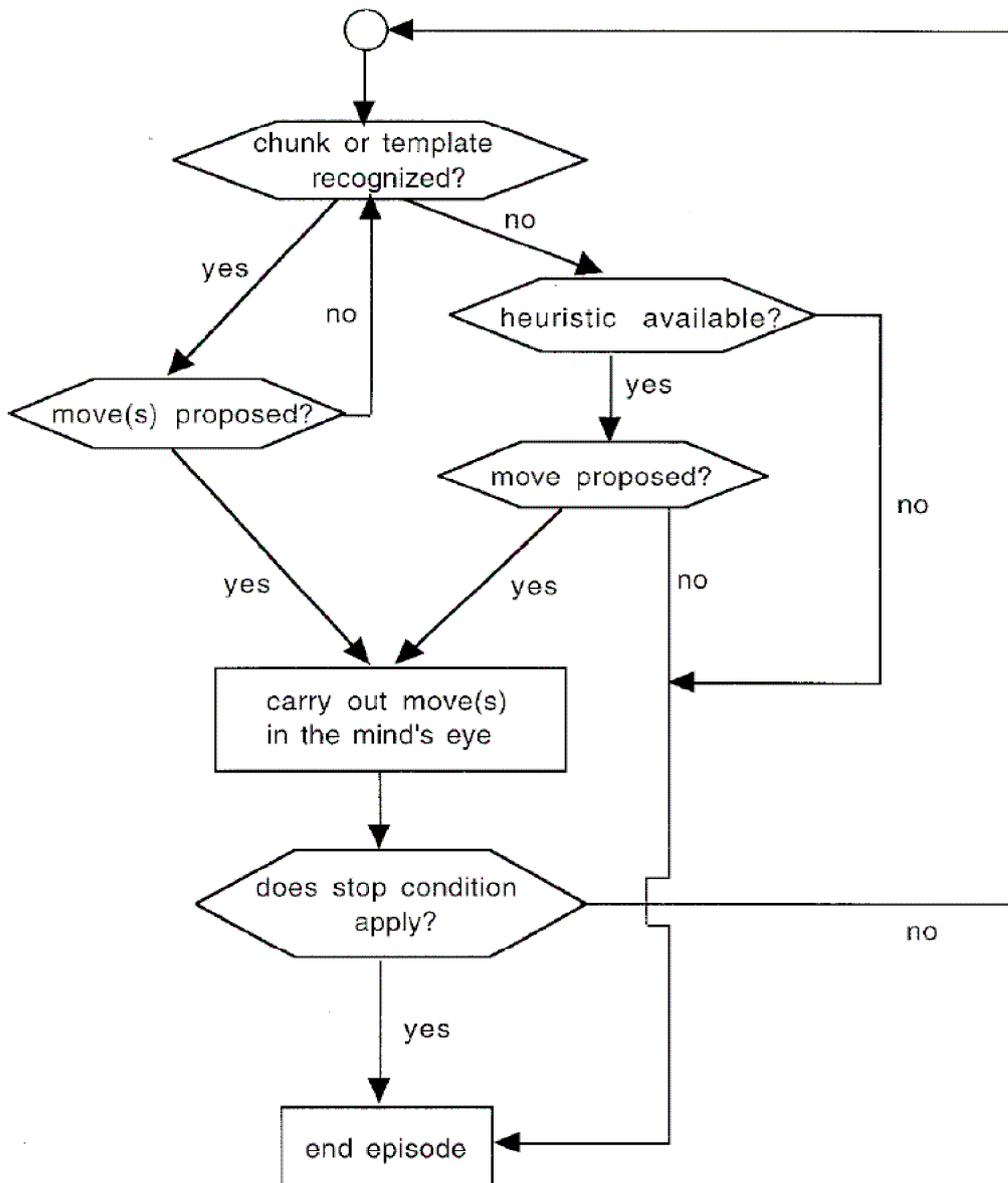


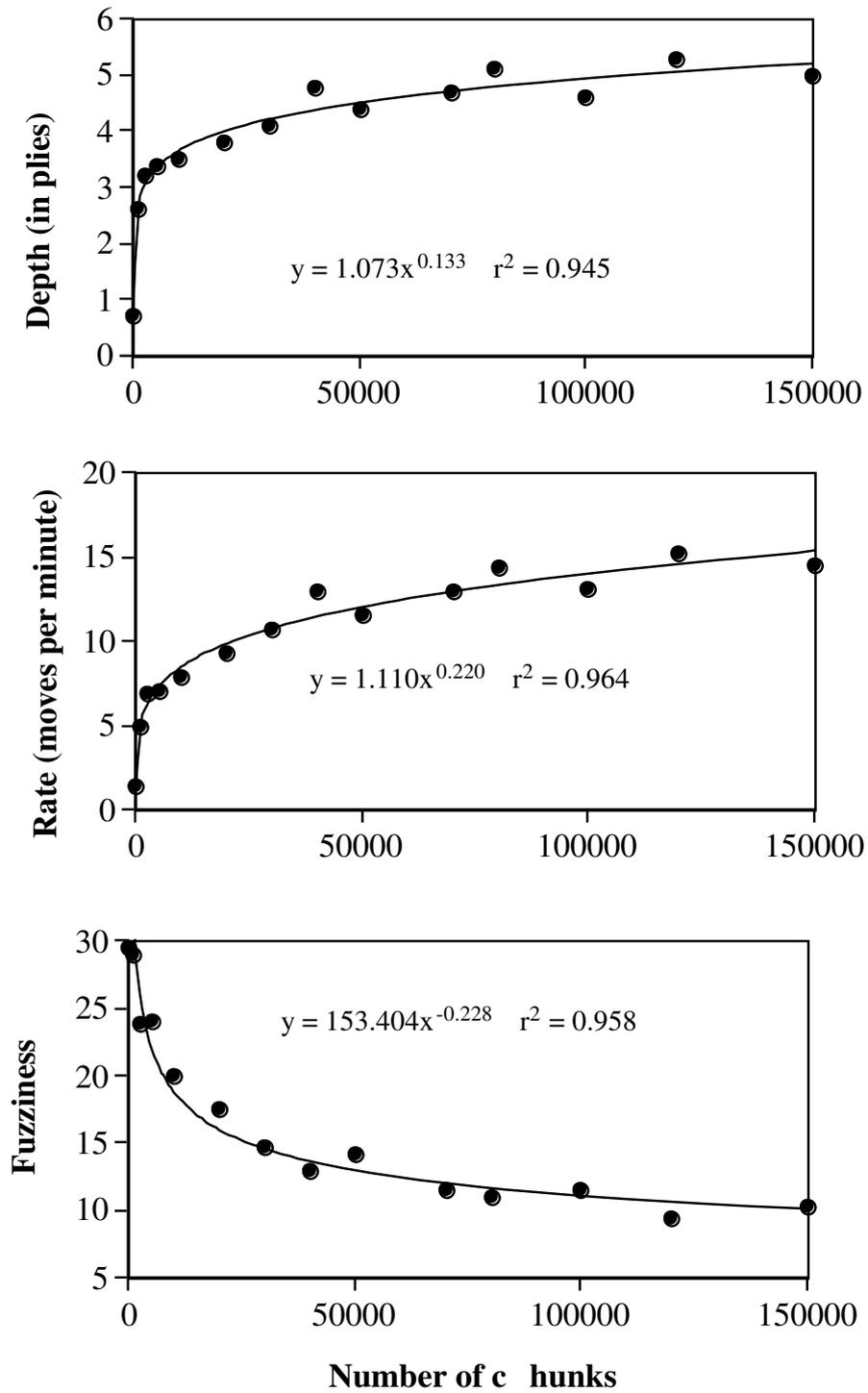
Figure 1

---

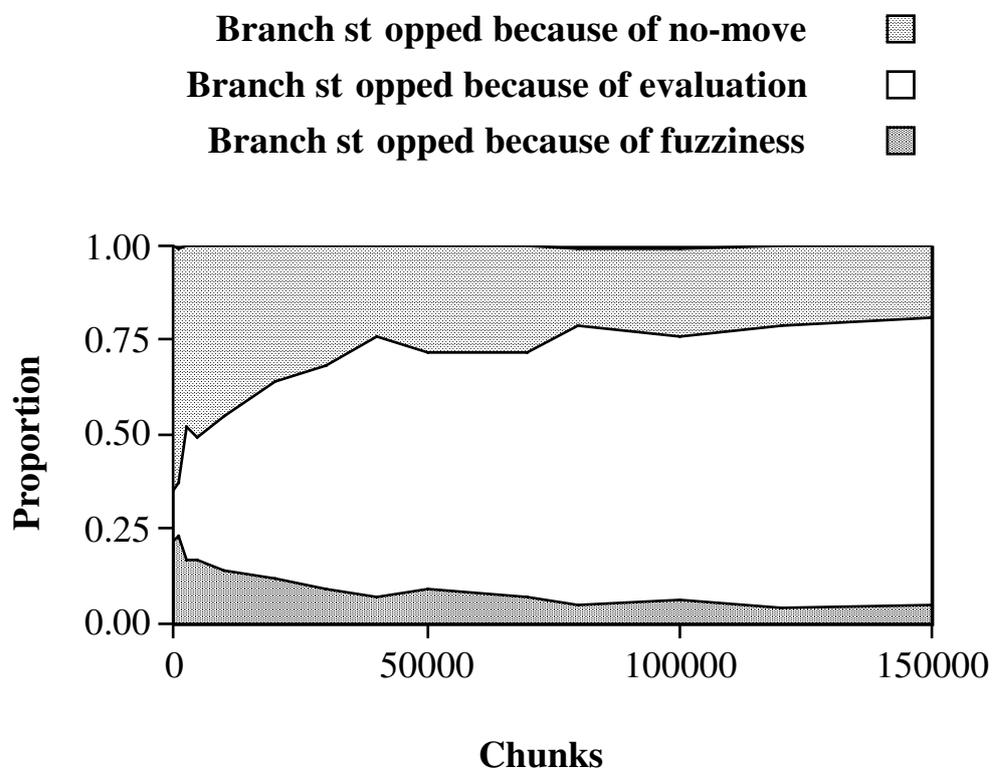
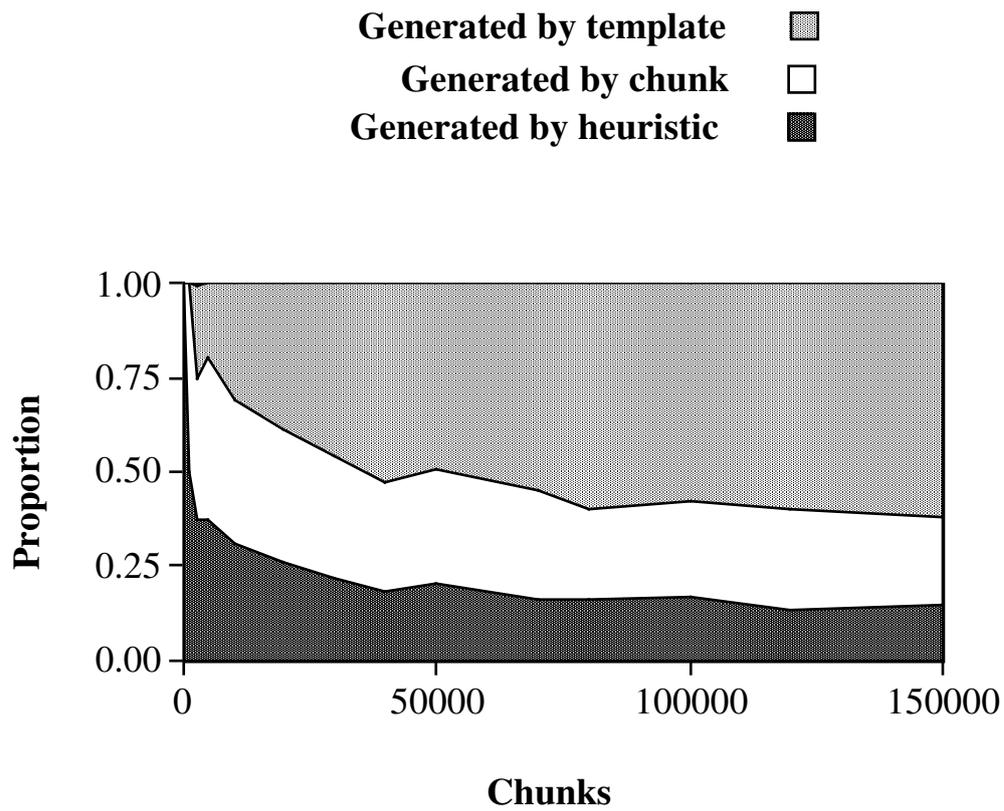
<b>Run #1</b>		<b>Run #2</b>	
? (one-search)		? (one-search)	
total time : 0		total time : 0	
depth of search : 0		depth of search : 0	
level of fuzziness: 0		level of fuzziness: 0	
CHUNK-FOUND		CHUNK-FOUND	
CHUNK-GENERATES-MOVE		HEURISTIC-GENERATES-MOVE-	
		MAYBE	
total time : 2		NO-MOVE-GENERATED	
depth of search : 1		SEARCH STOPPED	
level of fuzziness: 1			
CHUNK-FOUND		total time : 20	
CHUNK-GENERATES-MOVE		depth of search : 0	
		level of fuzziness: 0	
total time : 4			
depth of search : 2		<b>Run #3</b>	
level of fuzziness: 2		? (one-search)	
CHUNK-FOUND		total time : 0	
CHUNK-GENERATES-SEQUENCE		depth of search : 0	
		level of fuzziness: 0	
total time : 16		CHUNK-FOUND	
depth of search : 8		HEURISTIC-GENERATES-MOVE-	
level of fuzziness: 8		MAYBE	
TEMPLATE-FOUND		HEURISTIC-GENERATES-MOVE	
TEMPLATE-GENERATES-MOVE			
		total time : 20	
total time : 18		depth of search : 1	
depth of search : 9		level of fuzziness: 40	
level of fuzziness: 4		CHUNK-FOUND	
CHUNK-FOUND		HEURISTIC-GENERATES-MOVE-	
HEURISTIC-GENERATES-MOVE-		MAYBE	
MAYBE		HEURISTIC-GENERATES-MOVE	
HEURISTIC-GENERATES-MOVE		TOO-MUCH-FUZZY	
TEMPLATE-EVALUATION		SEARCH STOPPED	
SEARCH STOPPED			
		total time : 40	
total time : 38		depth of search : 2	
depth of search : 10		level of fuzziness: 80	
level of fuzziness: 44			

---

Figure 2



Figures 3



Figures 4

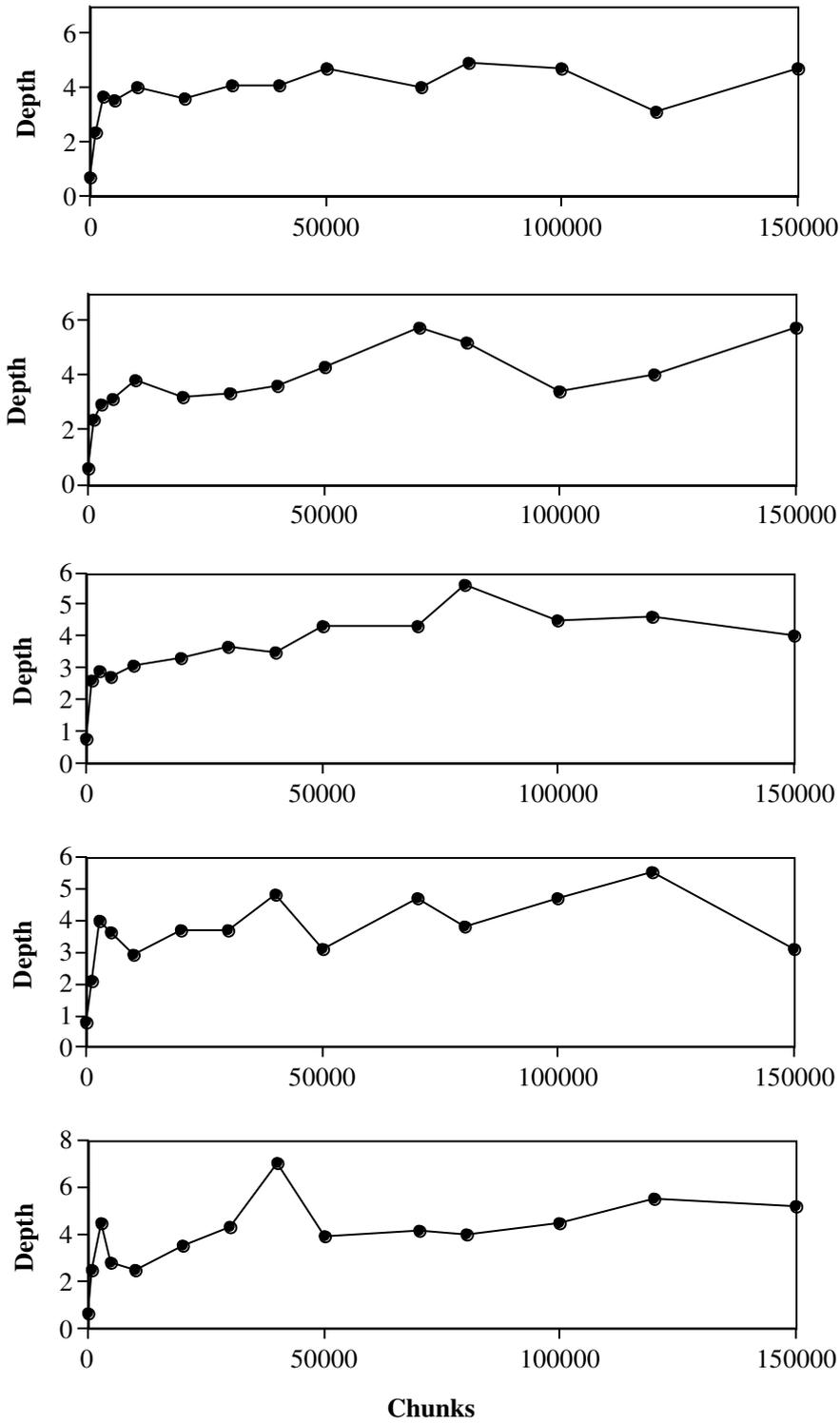


Figure 5

TABLE 1  
The Values of the 14 Nets Used in the Simulations

<i>Number of chunks</i>	<i>Number of templates</i>	<i>Proportion of templates</i>	<i>Probability of recognizing at least one template given a position</i>
0	0	.000	.000
1002	0	.000	.000
2680	5	.002	.180
5003	36	.007	.140
10011	100	.010	.240
20003	265	.013	.300
30001	493	.016	.360
40005	774	.019	.460
50006	1142	.023	.400
72044	2136	.030	.480
80010	2285	.029	.540
100000	2765	.028	.510
120000	3357	.028	.540
150000	4245	.028	.560

The values below 100,000 chunks were taken from simulations with CHEST, the others extrapolated. See text for details.

TABLE 2  
Heuristics, Chunks, Templates, and Their Relation to the Probability of Finding Moves and Evaluations

<i>Type</i>	<i>Heuristics</i>	<i>Chunks</i>	<i>Templates</i>
Probability of evoking an heuristic, a chunk, or a template	.95	$g(n\_chunks, .95)^a$	see Table 1
Probability of finding a move given that an heuristic, a chunk, or a template has been evoked	.50	$g(n\_chunks, .50)$	$g(n\_chunks, .90)$
Probability of finding an evaluation given that an heuristic, a chunk, or a template has been evoked	.15	$g(n\_chunks, .05)$	$g(n\_chunks, .50)$

<sup>a</sup>The function  $g$  is defined as:

$$g(n\_chunks, prob) = prob * 0.2 * \log_{10}(n\_chunks), \text{ if } n\_chunks > 0$$

$$\text{and}$$

$$g(n\_chunks, prob) = 0, \text{ if } n\_chunks = 0,$$

where  $n\_chunks$  is the number of chunks in a given net, and  $prob$  is a constant.

**TABLE 3**  
**Time and "Fuzziness" Parameters Used in the**  
**Simulations**

Time for carrying out a move	2 s
Time for computing move-generating heuristic	20 s
Time for computing evaluation	10 s
Upper limit of fuzziness	50
Increase of fuzziness per move	1
Increase of fuzziness per second	2
Decrease in fuzziness due to template	5

### Footnotes

<sup>1</sup>The ELO rating is an interval scale that ranks competition chess players. Its standard deviation (200 points) is often interpreted as a measure of skill class in chess. Grandmasters are normally rated above 2500, International Masters above 2400, Masters above 2200, and Experts above 2000. Below, classes (A, B, etc.) divide players by slices of 200 ELO points.