Air Traffic Control, chess playing and chess programs : lessons to learn

Jean-Marc Alliot, Jean-François Bosc

Air Traffic Control, chess playing and chess programs: lessons to learn

Jean-Marc Alliot Jean-François Bosc

16 octobre 1996

1 Introduction

Chess playing has been one of the first topics studied by psychologists. The first studies were conducted by Binet around 1894. He considered blind playing and the masters' internal representation of the chessboard. In 1925, a sovietic team including Diakov, Petrovsky and Roudik conducted a survey during Moscow tournament, where Lasker, Tartacover, Reti, Torre and others were competing. Later, Miller in the US, de Groot in the Netherlands, and Nicolaï Kroguious in the Soviet union, among the most famous, conducted similar works.

Moreover, the original goal of Artificial Intelligence was to model human reasoning. Since chess playing is one of the most prestigious symbols of human intelligence, it is not surprising that the founders of AI[1] very soon took active interest in chess playing (as soon as 1957). NSS, one of the very first AI computer programs, was a chess program. Since then, chess remained one of the most, if not the most commonly studied topics in AI. Some tremendous failures occurred, but some significant successes were obtained during the last ten years[2].

For these reasons, chess are one of the best known and analysed domains regarding as well player's psychology as reasoning modelisation and automation. Because of their similarities with Air Traffic Control, studying the results obtained about chess may provide useful lessons regarding psychology and training of the ATC Controller, modelisation of his reasoning, and automation of ATC[3].

2 Similarities and differences

2.1 Similarities

Air Traffic Control and chess playing share numerous likenesses.

Human individual and intellectual activity : A chess player and an ATC controller both perform a purely intellectual task. The player is alone while the controller works with a teammate.

Strong time constraint : In both cases the time constraint is critical. None can afford to be off delay. When a time crisis (which is called *zeitnot* in chess) occurs (because of a too complex position, or simply of past negligence), similar recovery techniques (that we will describe later) are used.

[1] Newell, Simon and Shaw, among others.

[2] Computer programs have now achieved Grand Master ranking, and in the last years the human world champion has been defeated twice by programs, once by Genius2 in a 20 minutes game and once by Deep Blue in a regular game([ABC+90]).

[3] Similar conclusions could be drawn for many other human activities which rely on the same pre-requisite.

Important theoretical and empirical expertise : In both cases training includes a theoretical part (rules of the game, theory of openings, basic tactical and strategic rules, ...in one case, air traffic rules and "systematic" conflict resolution rules in the other), and empirical rules which are quite difficult to define precisely (an IGM[4] can instantaneously "see" a combination, just like an experienced controller instantaneously solves a complex conflict, but they sometimes have difficulties to explain *how* the idea of the resolution came to their mind.

Short and long term predictions : Chess playing and Air Traffic Control require both tactical (short term, or "how to do it") and strategic (long term, or "what to do") predictions. The player needs some tactical sense to take profit immediately of favourable situations, but he also has to build a correct long term strategy. In ATC, the tactical and strategic parts are more clearly separated between the radar (tactical) and strategic controllers, but they successively switch positions.

Precisely defined workspace : In one case a "64 squares field" (Hans Ree), in the other a control sector defined by strips and radar screen.

Strong monitoring constraints : Both the controller and the player spend much time in monitoring tasks. The player needs to check that the tactical position doesn't become dangerous to make sure that he won't be defeated within a couple of moves while in a strategically winning position. The controller must ensure that the parameters that were valid when he established a resolution aren't changing, or reach the expected values.

Strong psychological constraint : A mistake has tragic and irreparable consequences : in chess it almost always causes a loss of the game, in ATC human lives may be involved.

There are probably many other similarities, but these are the most significant ones.

2.2 Differences

There are of course differences between ATC and chess playing. Some may be considered as minor, but others are more significant :

Predictable vs incompletely predictable universe : In chess playing the environment is entirely predictable, the player has total control of the situation. In ATC some parameters are out of control, either for technical reasons (uncertainties on speed and altitude, which are fuzzy values) or because of external factors (pilot mistake, technical failure, which are non-certain parameters). This difference is *critical*.

Playing against a human vs against the trafic : The ATC controller doesn't have a well-defined "opponent", therefore an important psychological dimension of the chess game is lacking, but this doesn't seem to be a very significant difference.

Different levels of consequences : Even though the psychological constraint is strong in both cases, it seems that the risk of killing people is more difficult to bear than the one of losing a game. However the difference is not as big as it appears in the first place. Some players virtually wager their live, and sometimes upset the keyboard or hit their opponent.

[4] International Grand Master

3 Psychological analysis

3.1 Reasoning processes

One of the most classical and famous studies of the reasoning process of a chess player was made by psychologist Adrian de Groot [Gro65].

He first tried to analyse the way chess players perceive a position. In this purpose he showed each subject a chessboard during 5 seconds, then removed the pieces and asked the players to put them back. He observed the number of correctly placed pieces and the method for putting them back, and noticed some interesting points :

- IGMs managed to put all the pieces back in almost 40% of the cases, while expert players never managed to do it even though they did perform pretty well.
- All chess players placed the pieces by logical groups, e.g. those attacking the castling and those defending it. Non-players operated without guidelines and obtained terrible results.
- The grand masters had a representation of the chessboard which was very close to an analysis. They memorised it as a set of attack and defense lines, not as a set of pieces.
- They used to rebuild the position through references to known similar past games[5]. They were able to reconstitute the main lines of the game that led to that position.
- When confronted to random positions (generated by placing pieces randomly) instead of positions resulting from real games, all subjects, either IGMs or non-players, obtained similar (and very bad) results.

Other studies were conducted to understand how "blind" players[6] mentally "see" the position. In a paper published in 1946 in the italian review *Minerva Medica*, Pina Morini considered that there were different types of memorisation, visual for some people and linguistic for others (who remember the sequence of past moves). On the other hand, analysing a position is always performed by logical sets and by analogy with known positions. It is thereby clear that the memorisation of a position is strongly linked with the knowledge of the domain, and is performed by connectionnist mental processes, which are very different from computer memorisation.

De Groot then studied the reasoning process of his subjects. He showed them chessboard positions and asked which move they would play, and why. He observed the following points :

- Most positions are first analysed by analogy with past situations. These analogies allow to determine moves which can provide some gain or avoid some loss, within a *global* long term strategy. One of the most striking examples was given by Kroguious ([Kro86]) :

  > "In the following position (figure 1), Bogolioubov found combination 22.♗xd5, exd5; 23.♖xg7+,♔xg7; 24.♕f6+, ♔g8; 25.♖g1+, ♕g4; 26.♖xg4+, fxg4; 25.f5 after only a few seconds.
  >
  > The explanation seems impossible to find out by referring only to general principles. The combination results from an analogy. Bogolioubov inconsciously made an association with a famous game, Bird - Morphy, London, 1858 (figure 2).

[5] *connectionnist* memorisation

[6] Blind playing consists in playing without seeing the chessboard. Some players are even capable of blind playing several games simultaneously. In 1947 polish grand master Najdorf played 42 games simultaneously, and obtained 36 wins, 4 draws, and only 2 defeats. His performance has been largely exceeded since then. Every good chess player is able to play blind.

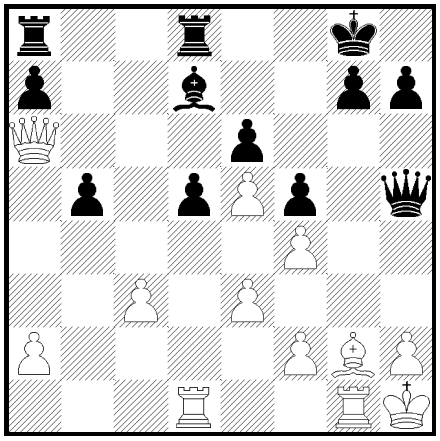

Figure 1: Reasoning by analogy

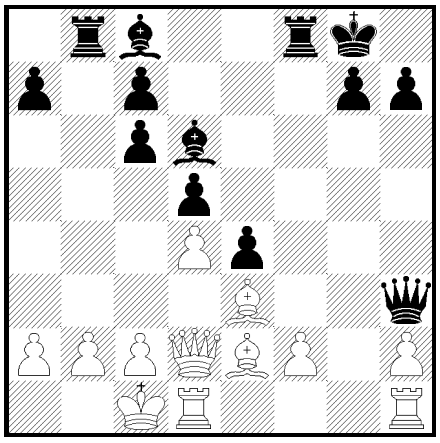

Figure 2: Bird - Morphy, London, 1858

The blacks played a winning attack 17...♖xf2; 18.♗xf2, ♕a3. Every experienced player can notice the similarity[7]."

- The player's search is highly selective. The main characteristics of grand masters is their ability to rapidly identify, by analogy, a few moves that need to be analysed precisely. All other moves are then ignored after a superficial analysis[8].
- The reasoning depth is limited. Even a grand master doesn't analyse any further than six or seven half-moves ahead (while the most powerful computers can evaluate positions up to fifteen or twenty half-moves ahead). This point can be linked with Richard Reti's joke : when asked "How many moves do you examine ?", he replied : "Only one, the good one !"
- The evaluation function is fuzzy : it is based on concepts which are difficult to define, e.g. *useful mobility*, *attack pressure*, or *active position* ... These criteria replace in-depth analysis, which is beyond the reach of the human mind. Evaluation is also often performed by referring to known patterns, which in some cases are easy to explain (central position of the towers, bishop of the appropriate color for a final) but sometimes are more fuzzy, or even depend on the player (for exemple Karpov likes to play the finals with knights).
- The player may change his strategy according to some analysis he made. He then has to reconsider the moves which require precise analysis.

De Groot's studies on chess playing are particularly interesting because they cover the reasoning techniques of humans for most of the problems they may have to face : the knowledge base is fundamental, and resolution is performed by referring to known examples[9]. This is made possible by the human brain's ability to organise information and almost instantaneously retrieve all the data relative to a specific object, provided that a precise identifier of that object is given. It is even more remarkable that some associations may remain after the corresponding data have disappeared[10]. One must take into account the enormous amount of data stored by a human brain during its life, and also the fact that the ability to memorise information is not the most important part. What is really important is the ability to efficiently recover the useful information when confronted to a few clues. Teachers know that even though understanding is important, some students won't be able to apply a well-understood method to a problem only slightly different from the one they learned.[11]. In this case the technique has been memorised, but the mind is unable to establish an analogy between the two situations. It is for the moment impossible to model[12] the way how the brain detects analogies[13]. If this was possible, we could not only build computers that would reproduce human reasoning, but also make humans more intelligent[14] by improving their ability to reason correctly[15]

[7] For those who wouldn't, the common mechanism is called a deviation : in both cases a piece is blocking the queen's path, and is deviated by giving up a figure.

[8] Which sometimes causes terrible mistakes, even among the greatest players

[9] There is a well known joke of a student how to heat hot water, when you know how to heat cold water: just let the water cool down, and solve the previous problem.

[10] For example some words, smells, or images, may initiate a feeling of "déjà vu", but it is often impossible to remember what these associations refer to.

[11] A math teacher even claimed that the exam subjects he gave to his students were always the same problem with some differences in presentation, and that there were always a few students unable to solve it after one year.

[12] With the possible exception of formal neural networks, but the model is still quite limited.

[13] Our personal view, following Herbert Dreyfus, is that this makes any attempt to create a computer program that would reproduce human reasoning unrealistic.

[14] Which justifies Patrick Winston's statement : Making computers intelligent may help making people more intelligent.

[15] We however fear that the brain's ability to establish analogies belongs more to the field of biology than to

3.2 Levels of expertise

Using De Groot's work, and some experiments conducted with GIM and computer programmer Julio Kaplan, Herbert Dreyfus tried to extract different levels of human expertise ([Dre87]) :

The beginner : He can only use some fixed patterns and totally deterministic rules which are independent from the context, and therefore easy to model on a computer. For example, in chess the patterns are the rules of the game (moves, captures, checkmate) and the values assigned to the pieces. The rules are "if ...then" sequences, e.g. "In an exchange, *if* the total value of the pieces that you loose is less than the total value of the pieces that your opponent looses, *then* do the exchange". Since a computer is much faster than a human player for this kind of task, it will easily defeat a beginner.

The experienced player : At this level, the player is able to identify similar situations and react by analogy with some past experience. He can identify a weakened castling or an under-developped center, which are notions difficult to express algorithmically. This kind of expertise can only be transmitted through examples. Anybody who has a significant practice of chess playing knows that the best books are gatherings of examples illustrating a particular theme, and that one of the main way to make progress is to study real games (either his own or famous ones), rather than learning rules in books[16].

The skilled player : At this level the player is acquiring the notion of game plan, or *strategy*. Developping a strategy is difficult, but the player is beginning to understand how to organize his moves to achieve the plan, and in what kind of situation a particular strategy is appropriate. The notion of plan seems very difficult to implement on a computer[17].

The master : At this level, the player can automatically sort out useless moves and plans that don't stand a chance of success. This process is unconscious, or at least impossible to describe.

The expert : An expert, typically an IGM in chess, doesn't consider a chessboard as a set of pieces in a play area, but rather as a set of possibilities inside which he is moving. At this level he can react automatically to a situation without any conscious reasoning and loss of time.

3.3 Mental representation

We will now describe three typical examples of human reasoning in a given position, depending on the player's perception of that position.

3.3.1 Inert image

The *inert image* of a position appears in relatively stable winning or riskless situations. "It's an evaluation of a position, viewed as a final judgement" (Kroguious). This image then becomes a default representation, and the player (or operator) doesn't reconsider the corresponding problems any more. This mechanism reduces the operator workload. It is one of the differences between a human expert and an automatic chess player.

---

the one of computers or psychological models. We also fear that the mechanisms of connectionnist memorisation can't be described by storage algorithms.

[16] For those interested in that kind of book, we recommend [Pac80].

[17] We're talking about strategic plans. There are interesting attempts to implement some notions of tactical planning.

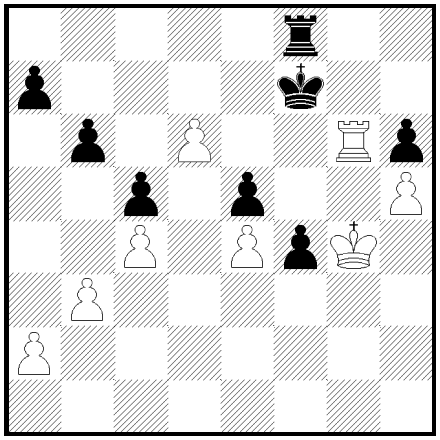

Figure 3: Petrossian - Kortchnoï, Moscow 1963

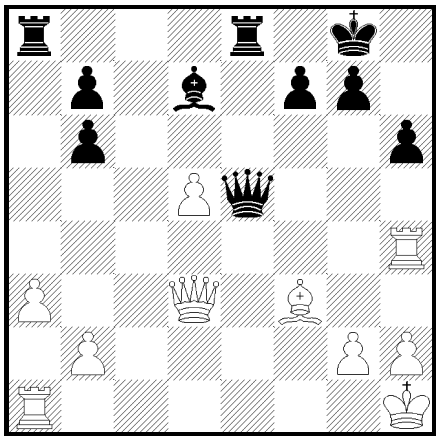

Figure 4: Mikenas - Bronstein

However such a mechanism may become very dangerous when the operator doesn't correctly keep watch over some parameters which guarantee that his image remains valid. One of the most classical examples is the position on figure 3.

After the game (that he lost), Petrossian said : "I was sure to win, the black's position was almost hopeless. Then I played 35.♖xh6??. I hadn't even imagined 35...f3!. If a good player doesn't notice such a threat, he would not see it even after half an hour analysis". In this case Petrossian didn't extract in his analysis the threat constituted by pawn f4, and the need to watch it carefully.

This kind of mistake is especially dangerous because it appears in situations where all risks seem to be under control. It is generally caused by an excess of self-confidence. Nearmisses caused by similar reasons are common in ATC.

3.3.2 Remaining image

It consists in transferring a past image to a current position. This mechanism often appears when realizing a delayed combination or resolution plan. It is typically human, and is not well modeled by computer programs. Figure 4 shows a typical example of a delayed plan.

Bronstein had been trying to checkmate in the first row during several moves. "I had considered ♖xa3 several times, but until then Mikenas always had a sufficient defense. Here he just had to play 25.♕d4 or25.♖e4 in order to hold the position, but he made the fatal

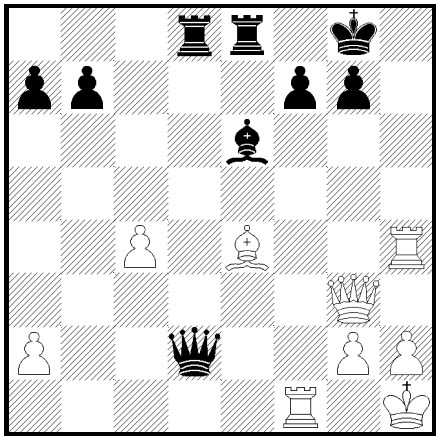

Figure 5: Tchekhover - Model, Leningrad 1933

error 25.♖b4".

Bronstein instantaneously played 25...♖xa3, and Mikenas had to give up. An ATC controller often uses this kind of mechanism : he plans a resolution well in advance, and implements it later on appropriate time.

However, this kind of reasoning also has some dangers. "The past may induce us to forget the current reality" (Kroguious). Figure 5 shows such an example.

Tchekhover said : "I noticed that it was impossible to play 29.♗h7+, ♔f8; 30.♕a3+ because of 30...♕d6; 31.c5, ♕e7. I therefore played 29.h3 to avoid a corridor checkmate. Model then made a mistake 29...♕e2?. After that 30.♗h7+ was a winning move. But I had removed it from my analysis, and was unable to take it into account again."

The difference between Bronstein's and Tchekhover's reaction is that the former had clearly identified the critical element to watch (the defense of a3 and the last row), while the other memorised the position as being final.

This kind of mistake, which consists in relying on a previously established analysis, may well happen in ATC.

3.3.3 Anticipated image

The anticipated image consists in seeing in a current position future elements which may make it turn into a winning or loosing position, in order to immediately take measures to improve the chances of winning or reduce those of losing. The elements on which this kind of analysis is based are sometimes very difficult to extract. Let us consider figure 6 :

Smyslov said : "I couldn't think of a way for the whites to win, and capturing the castle on h5 was tempting. However, my vision of the position didn't seem sound, so I chose 19...dxc4". After the game, the players realized that Smyslov had correctly anticipated the position (without being able to determine it). A precise analysis shows that after 19...gxh5; 20.♕xh5, ♖e8 the whites can play a4, and then ♗a3 which leaves the black king no chance of survival. In ATC it is probably very common to suppress a conflict by anticipation because of an uncomfortable feeling of a situation.

This ability to anticipate threats also has some negative aspects. "Sometimes ideas coming straight from your imagination may override reality" (Bloumenfeld). Thus, in the situation on figure 7, Flohr, worried by a possible move forward of the white pawn f, tried to prevent it. He played 28...f5. Unfortunately this reaction was fatal for him after 29.exf6, ♘xf6; 30.f5, g5; 31.♘h5, ♘xh5; 32.♕xh5. He shouldn't have overestimated future threats, and should have played 28...♖e8.

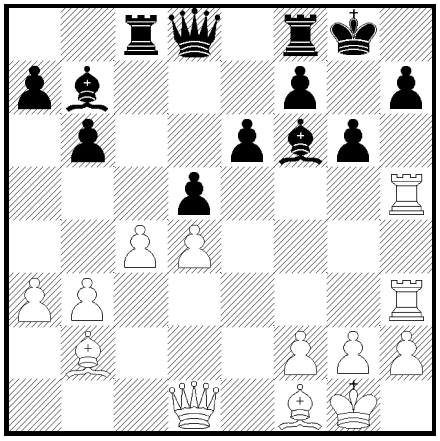

Figure 6: Keres - Smyslov, Zurich 1953 (blacks playing)

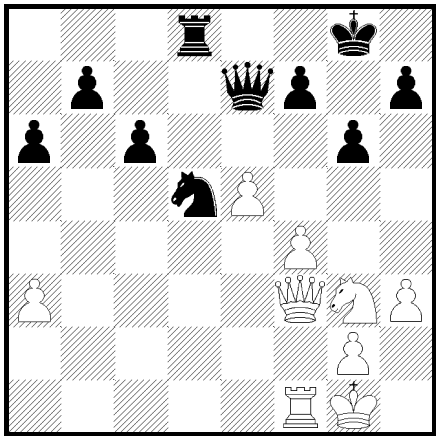

Figure 7: Bondarevsky - Flohr, Stockholm Interzonal, 1948 (blacks playing)

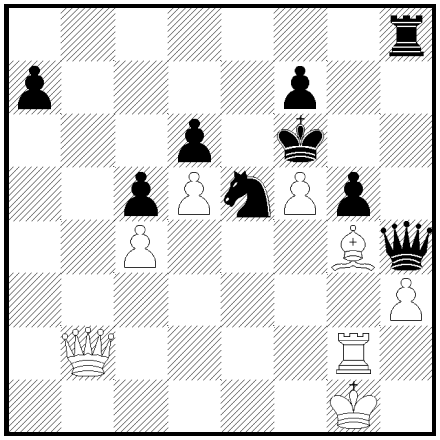

Figure 8: Romanovsky - Kasparian, Leningrad 1938 (black playing)

In ATC this kind of attitude consists in anticipating a conflict which is only a possible one, thereby leading to more resolutions than necessary.

3.4 Attention

3.4.1 Focussing attention

To focus one's attention on a particular area consists in isolating some portion of the chessboard (or of a control sector), and disregarding the interactions between that area and the remaining space in order to reduce the complexity of the problem. This method is typical of human reasoning, and is of course systematically used by all chess players, and probably all controllers too.

However there are some risks, like disregarding too many pieces (or aircraft), which may induce serious mistakes. Figure 8 shows an example of that kind of mistake.

The whites are in a bad position. Kasparian tried to speed up a checkmate, and played 52.♕e1+ announcing checkmate in three moves (53.♔h2, ♖xh3+; 54.♗xh3, ♘f3???). Romanovsky explained that the knight was blocked by his queen, which Kasparian refused to believe during a few seconds. He had disregarded the queen on b2.

For a controller, this kind of mistake consists in solving a conflict without taking into account a third aircraft that may interfere with one of the two aircraft involved in the conflict, simply because for some reason he disregarded that aircraft when he planned the resolution.

3.4.2 Instability

Instability of the attention causes simultaneous generation of several plans that overlap each other and lead to a mistake. Let us consider figure 9 :

Illine-Jenevsky said : "I could choose between two plans; I could exploit the weak position of the black castle via 1.♗f1, ♖a4; 2.♕e2, threatening ♕b5 or 2.♗d3, followed by ♗c2 or ♕e2. I could also take profit of the weakness on d5 by 1.f4, g6; 2.g4 followed by f5. I got the two plans mixed in my mind, and I played 1.h3, which is incorrect because the white bishop can not reach g4 because of ♗xd5. The game went on with 1...b5; 2.f4. Having lost some precious time I went back to my first plan, but too late ! After 2...b4; 3.axb4, axb4; 4.♗f1, bxc3; 5.♗xc4, ♕xc4; 6.bxc3 ♗xh3 the game was lost."

This kind of reasoning is similar to the one of a controller who considers two resolutions for the same conflict, and finally gets mixed up between them.

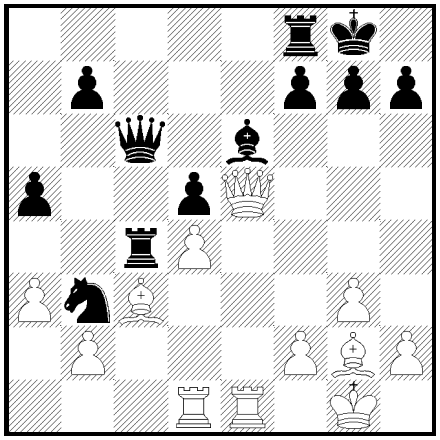

Figure 9: Illine-Jenevsky - Grigoriev, Moscow 1919

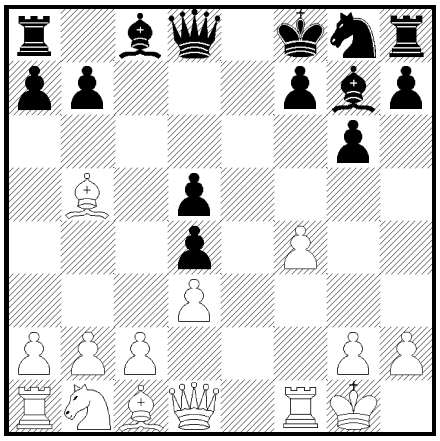

Figure 10: Alekhine - Blackburne, St Petersburg 1914

3.4.3 Lack of attention

In some cases a player focuses on his plan and doesn't pay sufficient attention to monitoring the game. There are some famous examples of such mistakes. Let us consider figure 10.

Alekhine completely focussed his attention on executing his plan in the center of the chessboard (deploy the pieces and attack the badly placed black king), and played 1.♘d2??. This enormous mistake (after 1...♕a5; 2.a4, a6 the bishop is lost) by one of the greatest geniuses of chess can only be understood by a total lack of tactical monitoring. Although this task is often tedious, as this example clearly shows it can't be avoided.

A similar problem occurs in ATC, where one can easily "forget" to watch the parameters of some aircraft, because there doesn't "seem" to be any problem, and the controller inconsciously prefers to deal with building a longer term strategic resolution, which is more rewarding[18]

3.5 The time crisis (Zeitnot)

Just like a chess player, the ATC controller is subject to strong time constraints. In chess it seems that some players are more prone than others to put themselves into zeitnot situations.

[18] Tactical monitoring and detction of tactical mistakes are probably among the tasks that would best be performed by computers. We'll come back to this point later.

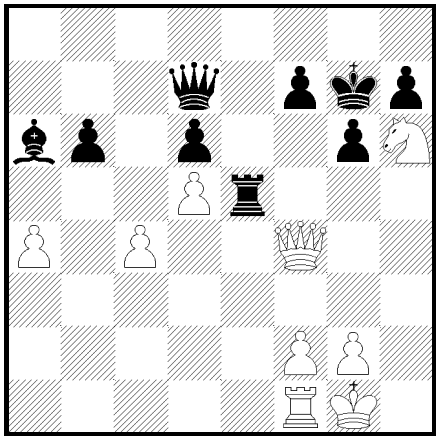

Figure 11: Kan - Flohr, Moscow 1936 (blacks playing)

Zeitnot exists in ATC too[19], and it would be interesting to know if the same phenomenon appears among controllers.

How does a chess player react in a zeitnot situation ? He first tries to simplify the position by realising as many exchanges as possible. An ATC controller reacts exactly the same way by suppressing as many conflicts as possible in order to reduce the number of those he has to keep watch on.

The chess player then plays the most evident and simple moves, just like the controller who will use the most classical and simple resolution techniques.

Although these changes of operating method are normal and even necessary, they may still be dangerous. Let us consider figure 11 :

Flohr had to play his fortieth move, just before time check. He was in a zeitnot situation. He therefore played the "obvious" move 40...♗xc4?? which apparently would bring in a pawn (he expected 41.♕xc4, ♔xh6). Kan played 41.♘g4! and Flohr gave up.

When confronted to a time crisis, a controller and a chess player use similar techniques, and are subject to similar mistakes : "obvious" moves which can be mistakes or create conflicts; attempts to simplify the situation, sometimes improperly, which may generate Brownian movements of aircraft ("This one bothers me, so I move it down; but then it conflicts with that one, so I move it down; but then ...")

4 Programming and automation

We showed that chess players and ATC controllers share a whole set of behaviours. We will now consider the evolution of chess programming during the last three decades, in order to examine whether the experience gained in that field can be useful in the one of ATC automation.

4.1 Historical review

4.1.1 "Strong" approach and cognitive modelling

The strong approach includes all attempts to develop computer programs based on exactly reproducing the human reasoning. In this framework, programs are "psychological theories

[19] and in many other domains : "Zeitnot is surprisingly similar to some critical situations observed in daily life" (Kroguious).

in themselves" (John Searle, [Sea87]). Newell, Simon and Shaw claimed to belong to that approach. In 1958, Simon wrote ([SN58]) : "Within 10 years from now,

- a computer program will be world champion in chess,
- a computer program will prove an important mathematical theorem,
- most psychological theories will be expressed as computer programs."

In 1956 Newell, Simon and Shaw had embarked on the LOGIC THEORIST (automatic theorem prover), and in 1957 on the development of two other programs : GPS (General Problem Solver) which was aimed at solving general problems using "human" reasoning techniques, and NSS, an automatic chess player. NSS was also supposed to use a human type of reasoning. Newell, Simon and Shaw expected to base their work on the results obtained by Miller and de Groot. A few papers were published, mentioning some "progress" of the program. In 1968, Scottish master David Levy launched a challenge : Newell, Simon and Shaw had bet that a program would be world champion at that time; Levy claimed he would take up the bet during the ten years to come ! In fact he was not taking a big risk. NSS could be considered a failure[20], and wouldn't have beaten a simple amateur.

In the meantime, in the field of cognitive psychology, verbalisation were looked upon as the keystone of the building of cognitive models.

In 1972, Herbert Dreyfus published a book ([Dre84]) that made a great fuss in the US. He brilliantly made fun of the works of Newell, Simon, Shaw, Minsky, etc. The controversial nature of the book initially caused negative reactions in the AI community, but, as time went by, his arguments had to be accepted. His criticisms were rewritten, restrained, and refined by several authors. John Searle[Sea90] brought them to the field of philosophy, Randall Davis applied them (without saying it) to the field of expert systems ([Dav82]), and even Terry Winograd was inspired by them.

In the field of cognitive psychology, the attackers were within the fortress. In 1977, Nisbett and Wilson [NW77] showed that the interpretation of verbalisations greatly depended on the conditions placed by the experimenter. On experiments based on the Hanoi towers problem, three different experiments ([GS62, Kar73, Eri75] gave different results: Gagne showed that verbalisation improves performance, Karpf that it had no influence on it and Ericsson that it had a negative influence. The only answer given by cognitive psychology is the following: if and only if the procedure is "executed" in verbal code, then verbalisation improves performance. Nisbett thinks that cognitive processi are not accessible by verbalisations, and that verbalisations are not the result of introspection, but of the use of "a priori" causal theories that subjects know because of their background. Caverni [Cav88] writes that it is impossible to build a cognitive model out of verbalisations by itself, and Ericsson and Simon [ES80] acknowledges the fact that an already built model has to be accepted before the interpretation of verbalisations. All this criticism can be related to Dreyfus, Searle and others works.

The publications written by Newell, Simon, Shaw, Minsky, and a few others during the sixties, being stupidly optimistic, greatly harmed AI, and especially the cognitive branch of AI. On the other hand, the criticisms raised by Dreyfus and those who followed him showed the great difficulty in making classical computers perform human types of processings[21], which require very different characteristics (ability to handle databases containing fuzzy and/or imprecise data, with instantaneous accesses probably based on connectionnist mechanisms

[20] just like GPS

[21] We do not claim that cognitive modelling is not an interesting field of study, but that computer programming must always be done by keeping in mind the structural differences between computers and human beings, and their respective strengths and weaknesses. This almost always leads to an appropriate (and brute-force based) approach for problem solving with computers.

rather than on algorithmic or rulebase reasoning). These criticisms favoured the development of different approaches (e.g. neural networks; Dreyfus and Searle admit that such approaches "may" give some results).

4.1.2 Learning

A review of game programming can't be complete without dealing with learning techniques. The first computer program using learning was a checkers[22] program written by Arthur Samuel (then with IBM). Samuel was a cautious researcher. He didn't publish any triumphant paper, but his work is of unquestionable quality. In particular, in his final paper he showed the advantages and drawbacks of classical learning techniques, and he suggested that they would especially be efficient on simple problems. Fifteen years later, a team from CMU[23] went back to his works and applied them to a simpler game, Othello. The program they developped, Bill, can be considered as one of the greatest successes in game programming. It defeated the US champion, and was probably close to the world top level. Many Othello playing programs are still using learning techniques. The current best program (Logistello) is using logistic regression to evaluate the coefficient of its evaluation function.

Most specialists consider that this method doesn't stand much chance of success in chess programming because of the high complexity of the game[24].

4.1.3 Algorithmic (or weak) approach

The algorithmic approach can be summed up by two concepts : computation speed and mathematical efficiency of search algorithms. In the beginning, the supporters of this approach were mocked at by the whole AI community, and also by the chess community. Pages could be filled up with jokes and treacherous comments made by prominent representatives of AI or chess (Simon, Minsky, Pachman, Reti, Kroguious,...)

However, as Richard Greenblatt, one of the defenders of algorithmics, noticed, the supporters of this method only needed faster computers and more efficient algorithms. They didn't ask for miracles or in-depth modifications of computer behaviour, as did the representatives of the cognitive approach.

History proved them right. During the seventies, Greenblatt's program, MacHack, began to bring some results. Algorithmic techniques improved (α-β, secondary search algorithm, SCOUT, SSS*, ...). In the early eighties, industry got involved in the problem. The first general public computers appeared. In the same time theoretical works kept on developping, and computing speed increased dramatically. In 1985, Feng Hsu had the idea to build a move generator plugged directly on a VLSI circuit. This was the beginning of the DEEP THOUGHT project. In 1989, DEEP THOUGHT put an end to twenty years humiliation of

[22] Checkers slightly differ from draughts, or international checkers (size of the draughtboard, captures, ...)

[23] Carnegie Mellon University

[24] Private communication by Robert Hyatt, one of the top researchers in the fiel of AI chess: "Several years ago (1970's) someone developed a program that used "patterns" to control its play. The patterns were general, ie don't allow an opponents rook on the same file with your queen and king with no intervening pieces, or you will lose your queen to a pin.... It was supposedly easy to enter patterns, the problem was that it was easy to develop patterns at first, but became very difficult later on, and the program never really got off of the ground.

A more recent program, called "morph" plays games against itself, and remembers patters in a similar way. It is "really stupid" and only learns by trying all sorts of garbage along the way. It *does* improve, rapidly at first, but then much more slowly.[...] I don't know where Morph is at present, but after a few years, it has yet to play in any computer event. I believe that the basic problem is that we simply don't understand how we learn patterns and how we match them mentally. Things like knight forks, bishop/rook skewers, pins, etc. are obvious to us after we get victimized a couple of times, and think about how many different circumstances you apply these "patterns" to. Writing an algorithm that does that is really *slow* since computers do not "think" as we do...."

game programmers by defeating David Levy 4-0 after a match during which it impressed all observers. DEEP THOUGHT had reached grand master level. Now renamed DEEP BLUE, and supported by IBM, it has defeated world champion Gary Kasparov in a regular game, a performance nobody could have dreamed of only a few years ago. Deep Blue evaluates millions positions per second, and in some cases may find a checkmate up to 15 moves (30 half-moves) ahead in tournament. Its tactical ability is extraordinary, and probably out of reach of any human player.

An other interesting problem in computer game playing is go. Currently, the best go programs are still very weak. There is still a debate about how to have a good program playing go. There are interesting works ([Che90, Bou95]), but there have been very little effort yet in that area. Presently, little is known about go evaluations, search strategies, etc. when compared with the amount of data available for computer chess programs and algorithms. When enough interest is generated to attract the same level of research to go as has been directed to chess (and now checkers), go will see much better programs. There is probably lot of work to do in mathematical modelling (especially using mathematical morphology) of the basic principles of go.

4.2 Man-machine cooperation

The cases of man-machine cooperation are quite seldom in chess. However, an interesting experiment was conducted by IGM Michael Valvo in 1989. Associated with computer program Belle he played against a team of human players including IGM Girome Bono. The experiment is precisely described in [Val90].

Here are some lessons he drew :

- The human player still largely surpasses the program for conceiving strategic plans.
- The program is extraordinarily efficient for finding tactical solutions.
- The program is extremely useful for avoiding mistakes, and allows the human player to focus on the creative part of the game by taking care of tactical monitoring.
- A pleasant technique consists in using the computer to check the tactical validity of strategic ideas.

After this general overview, we will now try to extract some useful lessons for ATC automation.

5 Conclusion

As Mike Valvo's experiment showed, a human operator is more efficient for the strategic parts, and therefore he needs to retain control of the distribution of tasks between himself and the computer, and of the general strategic plan. Thus, the computer could be used to relieve the human operator from some mainly tactical monitoring tasks.

We can take the argument even further : if we admit that air traffic control can be exactly modeled on chess game, the conclusion is clear and simple : we must focus on a purely algorithmic approach, and limit controller tasks to strategic planning.

However, there seems to be some differences between chess and ATC. In ATC the parameters are often fuzzy (speed and altitude are imprecise), and the problem is wider and more complex. Moreover, if the human operator has to remain the central element of the control process, a purely algorithmic approach can hardly be sufficient. As Donald Michie pointed out, if a human operator has to cooperate with a machine, he needs to trust it. Therefore

he needs to be able to understand every choice made by the computer. In particular Michie mentions ([Mic90]) the case of a chess program that played such a surprising move during a tournament that its programmer spent a whole night looking for a bug. The move was in fact correct, but was impossible to find for a human being without a long analysis of the position.

Therefore, if we want to keep a human operator in the process, we need to develop programs producing results that can immediately be understood by a controller, which, however, does not mean at all that they have to be built on a cognitive model; we firmly believe that it is more reasonable to expect computers to improve in speed and in performing tasks that they can already perform, rather than to start reasoning like we do.

We shouldn't delude ourselves; as Jonathan Schaeffer[25] said :

> *Although we in computer game playing have pretentions that most (some?) of our work is mainstream AI, it isn't. Despite all our efforts at knowledge engineering in Chinook, the program is still basically a brute-force searcher. Yes, there is lots of knowledge and it does help guide the search. But all this is secondary to the benefits we get from raw brute force computation.*
>
> *Regarding cognitive models of human beings: I am familiar with the theory that says the easiest way to build an intelligent machine is to model the human example. Personally, I think this approach is narrow- minded. I consider the human brain to be a machine, just as I consider my computer to be a machine. Both machines have strengths and weaknesses. My brain is very good at natural language and pattern recognition, for example. My computer is very good at repetitous tasks and performing arithmetic. On the other hand, my brain can only handle a small number of repetitious tasks before it becomes bored, and I have trouble adding a million numbers up in a second. My computer is not very good at either natural languge or pattern recognition.*
>
> *Given that each machine has different capabilities, it only makes sense (to me anyway!) that one should try and exploit the strengths of a machine and not its weaknesses. Programming a game, such as chess or checkers, one should take advantage of the computers ability to do mathematical calculations repetitiously. Hence, brute-force search is the natural outcome.*

We couldn't write a better conclusion.

Bibliographie

[ABC+90] Thomas Anantharaman, Mike Browne, Murray Campbell, Feng-Hsiung Hsu, and Andreas Nowatzyk. Deep thought. *Pour la science*, 156, Novembre 1990.

[Bou95] Bruno Bouzy. *Modlisation cognitive du jouer de Go.* Thèse de doctorat, Université Paris VI, 1995.

[Cav88] J.P. Caverni. La verbalisation comme source d'observables pour l'tude du fonctionnement cognitif. In *Psychologie cognitive, modles et mthodes*, 1988.

[Che90] K. Chen. Move decision process of go intellect. *Computer Go*, 14, 1990.

[Dav82] Randall Davis. Expert Systems: Where Are We? And Where Do We Go From Here? *The AI Magazine*, Printemps 1982.

[25] Private note : Jonathan Schaeffer and his team developped the checkers program Chinook that qualified for world championship final after ranking second at the US Open, the winner being the world champion Marion Tinsley (now deceased). Chinook was defeated by Tinsley in a head to head match.[SCT+92]

[Dre79] Hubert Dreyfus. *What computers can't do: the limits of artificial intelligence.* Harper and Row, 1979. ISBN: 0-06-090624-3.

[Dre84] Hubert Dreyfus. *Intelligence Artificielle, mythes et limites.* Flammarion, 1984. La traduction française correspond à la deuxième édition américaine [Dre79].

[Dre87] Hubert Dreyfus. Mind over machine. In Rainer Born, editor, *Artificial Intelligence, the case against.* Croom Helm, 1987. ISBN: 0-312-00439-7.

[Eri75] K.A. Ericsson. Instruction to verbalize as a means to study problem solving processes with the eight puzzle: a preliminary study. Technical Report 458, Stockholm University, Department of Technology, 1975.

[ES80] K.A. Ericsson and H.A. Simon. *Protocol analysis, verbal reports as data.* MIT Press, 1980.

[Gro65] Adrian De Groot. Thought and choice in chess. In Nico Frijda and Adrian De Groot, editors, *Otto Selz, his contribution to psychology.* Mouton, The Hague, 1965. ISBN: 902793438X.

[GS62] R.H. Gagne and E.C. Smith. A study of the effects of verbalization on problem solving. *Journal of experimental psychology*, 63, 1962.

[Kar73] D.A. Karpf. Thinking aloud in human discrimation learning. Technical Report 73-13625, University microfilms international, 1973.

[Kro86] Nicolai Kroguious. *La psychologie au jeu d'échecs.* Grasset - Europe Echecs, 1986. Recueil d'articles du psychologue et GMI Kroguious parus en 1967, 1968 et 1969.

[Mic90] Donald Michie. *Réflexions sur l'Intelligence des machines 25 ans de recherche.* Masson, 1990.

[NW77] R.E. Nisbett and T.D. Wilson. Telling more than we can know: verbal reports on mental processes. *Psychological review*, 84:231–259, 1977.

[Pac80] Ludek Pachman. *Théorie élémentaire du jeu d'échecs.* Payot, 1980.

[SCT+92] Jonathan Schaeffer, Joseph Culbertson, Norman Treloar, Brent Knight, Paul Lu, and Duane Szafron. A world championship caliber checkers program. *Artificial Intelligence*, 1992.

[Sea87] John Searle. Minds, brains and programs. In Rainer Born, editor, *Artificial Intelligence, the case against.* Croom Helm, 1987. ISBN: 0-312-00439-7.

[Sea90] John Searle. L'esprit est-il un programme d'ordinateur? *Pour la science*, 149, Mars 1990.

[SN58] Herbert Simon and Allen Newell. Heuristic problem solving: the next advance in operations research. *Operations Research*, Janvier 1958.

[Val90] Michael Valvo. Consulting chess with a computer. *International Computer Chess Association Journal*, 13(3), 1990.