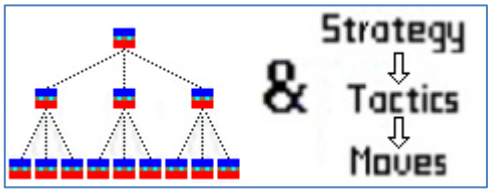
Starting points for improvement in Stratego programming



Summary

This document describes the current state of Stratego programming in literature and possible improvements for the status quo. It contains suggestions to implement some known programming techniques that already have been implemented in other areas of game programming. The arsenal of currently used universal algorithms is not sufficient to accomplish really significant improvements in the playing strength. Real progress can be made by the implementation of game knowledge or by algorithms that have not been used until now in Stratego programs. In this article attention goes to a first concept of structured theoretical game knowledge and thereby a framework for decision making in Stratego programs. From this framework suggestions are made about the ways game knowledge can be implemented and combined with current conventional algorithms.

Contents

Preface

Years ago I discovered that Stratego constitutes an ultimate challenge for programmers of artificial intelligence. When there was time enough to accept this challenge seriously I was able to test some items which were not mentioned in Stratego literature. The first results were disappointing. They showed that my ideas only covered a tiny part of the Stratego problem.

My conclusion was (and is) that only a broad theoretical base including game knowledge may offer a real perspective to better playing Stratego programs. Therefore I have tried to extend my theoretical knowledge of the game. That has led me to conclusions about what is necessary to make a Stratego program that will play on a reasonable level. Because preliminary research is required for most of the suggested improvements the realisation is going to take much, much time, probably many years.

The use of game knowledge appeared to be necessary. This has consequences for the chance that Stratego programming will be improved substantially.
The academic world prefers to improve board games by the use of universal applicable algorithms that do not depend on domain knowledge. In case of Stratego this preference may be reinforced by the lack of literature that in detail describes how to play Stratego. This leads to the question what efforts the academic world is willing to spend to the game of Stratego in the future. The preference for issues with a universal applicability leaves chances for only a few methods (probably Monte Carlo tree search or self-learning) that have not yet been used for Stratego. This tendency will hamper farther investments in the implementation of domain-dependent algorithms. Development by a commercial company is improbable because the return on investment is insufficient. Therefore progress by game knowledge will depend on the interest of individual persons in the improvement of the playing strength of Stratego programs. This document shows that a long way has to be gone, and the amount of work may be too voluminous for one person. For me it will take years to do research and develop a program. The dependence on individual effort is a reason for me to share knowledge about the game and programming techniques. In my view not sharing what should be common information is a waste of time.

This document is not a design for a Stratego program. It contains an inventory of points where improvements are possible in comparison with programs that have been described in current literature. The exclamation symbol  indicates the presence of a subject that is suitable for some kind of improvement. For most of the improvements preliminary research will be required.
Current literature does not fully reflect the current state of the art. However it is very well possible that some of the mentioned steps already have been fixed by someone else. Nevertheless I hope this paper will be a source of inspiration for anyone who has the same fascination for the challenge that Stratego offers.

Han Wolf, September 2018

1 Introduction

1.1 Status quo

More than 15 years ago – in May 1997 – a computer program defeated the world champion chess. That was a milestone in the development of programs for board games. Suddenly people recognized that more sooner than later humanity should bow for the superiority of the computer in other board games. In fact gradually the computer has been developed as a formidable opponent in most other board games.

But Stratego is one of the board games where the computer has not attained the level of human master players. In the academic world research has been done in order to improve the playing strength with known domain independent algorithms, but these attempts have led to not more than a mediocre playing strength. With the standard techniques described in current literature a master level in Stratego is unattainable. This leads to the more positive conclusion that a whole domain lies in front of us that offers more than enough challenge for phantasy and experiment.

1.2 The challenge

Why is it so difficult to make a good playing Stratego program? Most important reasons are:

- The initial position can be chosen completely in accordance with personal preference
- The ranks of pieces of the opponent stay unknown until they duel.

This lack of information makes Stratego more complex than board games with a fixed initial position and pieces with known ranks.

A practical problem too is the lack of literature with expert knowledge about Stratego. Well filled libraries are available for games like draughts, chess, go, etcetera. Anyone who wants to know how Stratego should be played can find some fragmentary and brief directives on Internet [VB, JM2][1]. In a tutorial of the Probe program more global directives with examples can be found [IS].

All this makes Stratego programming rather difficult, but there is more to it.
In Stratego gains or losses originate from duels between pieces. Before pieces come to a duel they have to bridge a distance and in Stratego usually this takes a lot of moves. So most of the moves in Stratego are moves to an empty square on the board. The value of a move to an empty square depends on the goal that is being pursued by the move, in most cases a duel that should be attained or avoided. So it is necessary to look ahead to a goal. In board games the look ahead usually is being achieved by a brute force analysis of a search tree. But most tactical goals in Stratego only can be detected by a look ahead of many more moves than practically is possible with a search tree.
A trustworthy evaluation of most moves in Stratego cannot be achieved by only a brute force analysis of a search tree, additional methods are necessary.

1.3 Dynamic and static look ahead

Literature draws a distinction between:

- Dynamic look ahead by an analysis of moves in a search tree
- Static look ahead by an analysis of characteristics in a game position.

In his thesis the author Vincent de Boer describes his program Invincible as a program that only uses static look ahead [VB]. A program based on agents by Mohannad Ismail [MI] too does not use search trees. But most programs in academic research use exclusively search trees for the determination of the best move in a game position. Static look ahead is essential for improvement of the playing strength of Stratego programs. Probably a combination of both dynamic and static look ahead offers best chances for improving the playing strength. A combination of search trees and static look ahead has been implemented by Imer Satz in Probe, the current world champion of Stratego programs.

[1] Chapter 9 contains a list of these literature references

2 Already used decision methods

For the academic world Stratego is of importance because it offers a challenge to the theory of artificial intelligence. Literature mentions various methods to let play Stratego by a program:

- Search trees
- Goals and plans
- Agents
- Monte Carlo
- Genetic algorithms
- Computer learning with convolutional neural networks.

2.1 Search trees

In most board games the authors use search trees where the best move is determined by some kind of minimax method [WP1]. In Stratego this principle is applicable too, but the search tree has to be adapted because unknown pieces of the opponent can have various ranks [SA]. If an unknown piece may have more than one rank then this enlarges the number of possibilities. This makes the search tree broader. The broader a search tree, the less the capacity to look ahead by brute force.

The academic world has especially given attention to diminish the size of the search tree in Stratego. All theoretical possibilities with regard to this aspect have been applied exhaustively [SA]. Restrictions to processing time limit the horizon of tree search in most of these studies to 6 moves or less. In Stratego it is necessary to look ahead much farther than 6 moves. In order to look ahead farther algorithms are necessary that look ahead without the execution of moves.
The author Imer Satz of the world champion program Probe has improved the look ahead in the program by analysis and evaluation of free paths in the nodes of the search tree. The future will learn whether more methods can be found that enlarge the look ahead in search trees in Stratego.
Who wants to learn about shortest path algorithms can find information on Wikipedia [WP6].

2.2 Plans

The program Invincible is one of the better Stratego programs [VB]. It chooses the best move from plans in an actual game position. This is a realisation of the principle “look ahead without the execution of moves”. The author Vincent de Boer has been world champion Stratego a number of times and thus has an extensive and deep expert knowledge of the game. Apparently the possession of such knowledge is a necessary condition for the development of a program that works in accordance with this principle.

2.3 Agents

The use of agents is a method to distribute a complex problem over more than one problem solver. Mohannad Ismail is the author of a Stratego program that has been based upon the interaction of agents [MI]. Each agent represents a piece that in a specific way looks at the position on the board, evaluates situations and complies with specific rules of behaviour. The author assumes that expert knowledge of Stratego is necessary to make a strong playing program with this method.

2.4 Monte Carlo

This method tries to determine the best move in a game position by simulation of game positions. A random sample of game positions is generated. In generated game positions moves are done and evaluated in a simple way. Information about the current game is used both in the selection of positions and the selection of moves. A move is chosen that occurs as the best move in most game positions of the sample.
Jeroen Mets has done a study of Monte Carlo Stratego [JM1] and concludes that Monte Carlo with random generated positions and random generated moves gives a slight improvement of the playing strength in Stratego.

The Monte Carlo method has been mentioned in an article of Jeff Putkin and Ju Tan [PT] too.

2.5 Genetic algorithms

The use of genetic algorithms is a method to optimize evaluation functions of Stratego programs. Weight factors of criteria in an evaluation function are varied and the resulting evaluation functions are used in games played against opponents with a fixed playing strength. This approach is a search for the combination of weight factor values that produces most wins in games. The success of this method depends on the set of evaluation criteria that should be optimised by weight factors.
A study by Ryan Albarelli, and more recently a study by Vincent Turu and R.M. de Boer, has shown that the optimising of parameters in an evaluation function is possible and really leads to improvement of the playing strength.

2.6 Computer learning with convolutional neural networks

Artificial neural networks are being used for machine learning, as applied to speech and object recognition, image segmentation, modelling language and human motion, etc. Recently a study to learn play Stratego by convolutional neural networks has been described in an article of Schuyler Smith[SS]. A slight improvement with regard to a basic playing strength has been achieved.

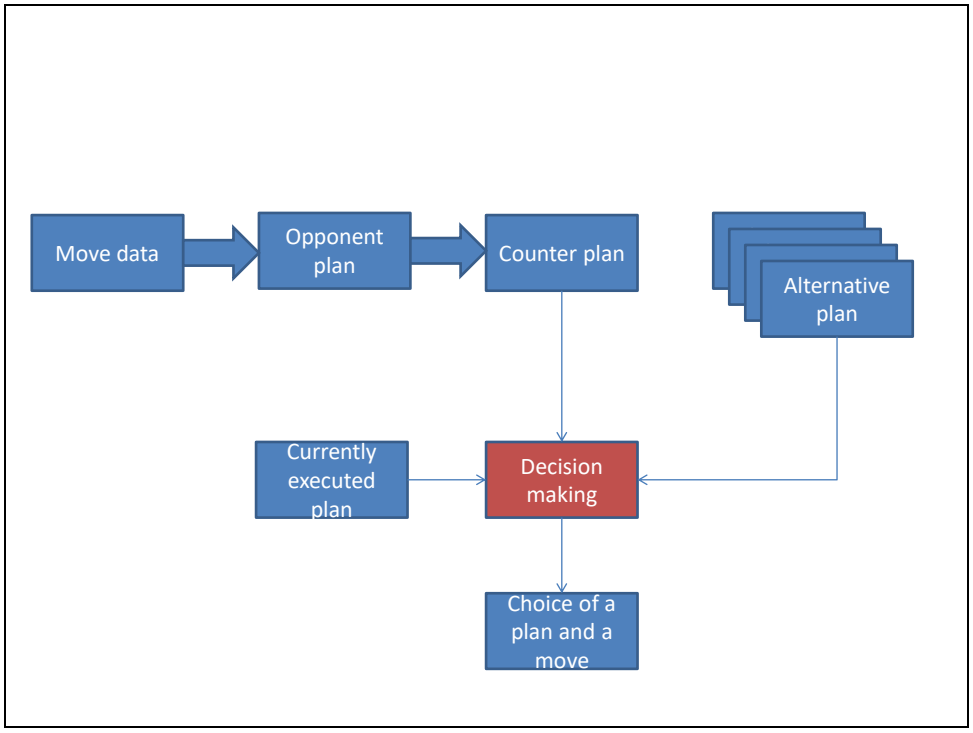
3 The decision process in a Stratego program

3.1 Decision making by human players

Uncertainty and lack of information in Stratego require a kind of decision making that differs from decision making in games where decisions primarily are based on exact information. A practically effective approach is to apply structure to the own actions in order to get more hold on situations where anything is uncertain.

Literature about expert knowledge of other games often distinguishes strategical and tactical levels of decision making. Do human players apply consciously these levels in their thought processes? Only guesses are possible here. No research is available, therefore here a first and speculative attempt.

It seems that the focus of a human player is on plans; probably these are the most used “units of thought”. On that base a player chooses a move.
A schematic presentation of this process may look like:



A human player selects a plan from a number of candidate plans and keeps to this plan if possible.

It may be necessary to interrupt the execution of the current plan, for example if the opponent creates a threat and a counter plan is required.

It is possible to switch between to different plans in order to confuse the opponent. It too is possible that new information may lead to a preference for alternative plans. Much variance and flexibility may be present in the approach of a human player. Besides plan orientation and technical competence human players let themselves guide by categories like “be unpredictable”, “apply bluffs” and “observe the opponent and use psychological opportunities”.

Is it possible to let the computer think like a human being? At the current state of the art this does not seem achievable; other ways of decision making are required in a Stratego program.

3.2 Decision making by a program

A computer program can only work with exact rules and quantities. Therefore decision making in a Stratego program is model based and rigid.

Literature about expert knowledge in other games often distinguishes strategical and tactical levels of decision making. This division into levels of decision making may also be a valid approach for a model based approach of Stratego. So here three levels of decision making are distinguished:

- Strategic decision making
  On a global level the player makes choices which influence will be felt over a period of tens or even hundreds of moves. These choices restrict possible choices on a lower level of decision making.
- Tactical decision making
  The player recognises goals within the framework of the strategy. An analysis of goals shows what their relative value is and what moves may attain these goals.
- The choice of moves in an actual game position
  The player determines which moves fit within the framework of strategy and tactics. Only moves that fulfil these criteria are relevant and lead to the choice of a move in the current game position.

What's written here about strategy and tactics is not really original. But a rather strange fact is that in academic literature about Stratego programming little or no explicit attention has been given to strategy or tactics as a factor of influence for the choice of moves. The word "strategy" is used in a diffuse way and descriptions of game knowledge are only global and fragmentary. Even in an article about the using of domain-dependent knowledge in Stratego [JM2] a division in strategy, tactics and move choice is absent. Because so little research has been done to these subjects this is an interesting area for improving the playing strength.

3.3 Decision making by rules or domain-independent methods

The human decision process is based on systematic thinking according to plans. From knowledge and experience a human player makes choices within that framework. This is a complex thought process that is hard to express into rules. In computer programs it is possible to avoid the complexity of this thought process by the replacement of this thought process by the brute force analysis of a search tree or other methods. Methods that are only rule based have a serious handicap. Their decision process only can be an incomplete representation of the human thought process. Even for a strongly reduced representation it is a heavy task to produce survey-able and maintainable program code and data structures. Therefore it is advisable to apply – wherever possible and meaningful – a decision process by domain-independent methods.

3.4 The choice of an initial setup

Preceding to the game the program chooses a setup that is sufficiently playable and unpredictable. Eventually the program takes into account what kind of position the opponent has used in the past.

There exists some literature about the generation of setups [VB], but the descriptions do not contain sufficiently detailed information for the development of an algorithm. Farther research in this area is necessary. The Gravon database contains a large number of setups and games, but a classification of the available setups is necessary for the accessibility of positions in this database. Maybe it is possible to develop algorithms that generate sufficiently playable and unpredictable setups from statistical data in the Gravon database.

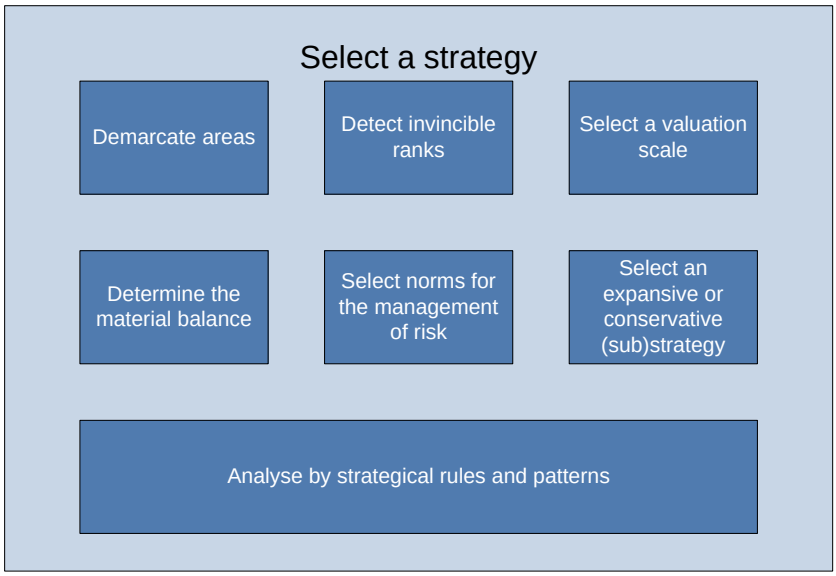
4 Long term decision making: strategic choices

A goal of strategy is to enforce long term coherence between moves. The player makes choices that have influence on the choice of moves during a period of tens or hundreds of moves. Strategy cannot be determined by computing but is based on human knowledge and experience. So programming of strategy requires game knowledge. In literature only some general and fragmentary remarks are available about strategy. This document contains a first attempt to define a framework for strategic game knowledge. What is written here probably is not the ultimate truth about strategy in Stratego, but somebody has to make a starting point. So here are presented characteristics that the author of this document has learned by a study of Stratego games in the Gravon database and by playing (mostly for fun) against Stratego programs.

In Stratego strategic decision making refers to:

- Areas
- Invincible ranks
- Valuation of ranks
- Ranks and probabilities for unknown pieces
- Distribution of most important ranks over the board
- Material balance
- Control of risk
- Conservative or expansive approach.

The following scheme shows the main functions that play a role in the determination of a strategy.



4.1 Areas

Three lanes exist between the lakes. Corresponding to these lanes three areas exist where pieces can move. In the initial setup these areas consist of four empty squares, but as pieces leave the board the size of these areas grows.
Pieces in one area do not have influence on pieces in a different area. This enables the choice of a local strategy for each area. As the game progresses more open connections arise and finally this ends in a common strategy for the whole board.
The choice of a local strategy cannot be done by tree search but depends on human knowledge of the game. Somehow rules and their corresponding actions have to be recorded in a Stratego program.

4.2 Invincible ranks

A rank is invincible if the opponent has no ranks to conquer a piece with the invincible rank; at most an exchange is possible. Invincibility arises for the marshal if the spy of the opponent disappears from the board. Generals become invincible if the marshals are exchanged.
A rank can be locally invincible if the opponent has no higher rank in the vicinity.
The presence of invincible ranks has influence on all facets of strategy and tactics. All decision making about strategy and tactics should differentiate with regard to this aspect.

4.3 Valuation of ranks

The method of valuation of ranks has a direct influence on various processes:

- The determination of the material balance
- Risk management
- The evaluation of duels.

The playing strength of a Stratego program is being determined partly by these processes and thus by the valuation method too.

Literature mentions a number of different valuation scales with fixed values for ranks. But these value components do not have the same meaning in all articles; their value and their ratio differ. Authors of these valuation scales define specific corrections for conditions that may occur in game positions.

In order to make the valuation scales of different authors comparable in the following scheme a considerable amount of the nuances in the original scales have been omitted.

Rank	JM	KS	MS	SA	VB	Study
Marshal	100	100	100	100	100	100
General	50	90	75	50	86	95
Colonel	20	58	44	25	59	77
Major	10	37	35	19	41	61
Captain	4	26	25	12½	28	50
Lieutenant	2	21	12½	6¼	20	42
Sergeant	1	16	5	4	13	39
Miner	10	32	25	6¼	9	39
Scout	½	18	2½	7½	6	33
Spy	20	42	25	50	50	95
Bomb	150	26	19	5	50	42
Flag	200	316	250	2500	100	-

All scales show a gradual decrease of the rank values from marshal to sergeant. This trend is not present in the ranks of miner, scout, spy, bomb and flag.

The last column contains the results of a statistical study of the Gravon database by the author of this discussion. The base for these values is the material gain that has been achieved per rank in about a million duels of about 29.000 games. The value of the flag depends on the method of decision making in a program and therefore has not been recorded in the last column.

The valuation scale has a relevant influence on the quality of various crucial decisions that a Stratego program makes. The quality of the decision process is the final measure for the quality of a valuation scale. The influence of the values in a valuation scale should be traceable. That requires extra facilities in the program. Only then it is possible to determine by trial and error which valuation scale is effective.

Most authors have the opinion that eventually values of ranks can change in accordance with certain conditions that can occur during a game.

Here follow some examples of conditions that may alter the basic value of a rank:

- An increase of the value if the number of available pieces decreases
  This tendency has been mentioned by Vincent de Boer [VB] in his thesis.
  In particular this kind of condition is applicable to ranks with special abilities like miners and scouts.
- A high rank plus spy may offset a marshal
  The exchange of a marshal against a general may be favourable if the marshal of the opponent is known and the own general is unattainable for the marshal of the opponent. The locally invincible general may conquer material and the marshal has to beware of the spy. But other conditions favour a non-capture of the general by the marshal.
- If a bomb blocks access to the flag then it has a large value
  But when the rank of a bomb has been exposed and the bomb does not block access to a relevant area or valuable pieces then its value is almost zero. The capture of such bombs by a miner has no added value at all.

Many more examples may be given. A separate document “Static value of ranks in Stratego” contains a more elaborate survey of basic values and value variations. This shows that it will be necessary to implement dependencies on more and other conditions than have been mentioned by authors in current literature.

The rules for the valuation of ranks have to be recorded somehow in a Stratego program. This may be done by the use of:

- Hard coding in the program
- Decision tables
- An expert system.

Probably the use of decision tables is a good choice, because as insight in real valuation grows it will be necessary to change and / or extend the rules. The number of rules is too small to require the use of an expert system.

4.4 Ranks and probabilities for unknown pieces

4.4.1 Positions on the board have various probabilities

At the start of a game the ranks of the pieces of the opponent are unknown. Theoretically the probability distribution of ranks is the same for every piece and square.

Rank	Number	Probability
Marshal	1	0,0250
General	1	0,0250
Colonel	2	0,0500
Major	3	0,0750
Captain	4	0,1000
Lieutenant	4	0,1000
Sergeant	4	0,1000
Miner	5	0,1250
Scout	8	0,2000
Spy	1	0,0250
Bomb	6	0,1500
Flag	1	0,0250

The Gravon database contains a large number of Stratego games. A study of the setups shows that the probability distribution of the ranks for an unknown piece is different for each square on the board. That's no surprise. Human players do not place pieces at random in the setup but give positions to the pieces in accordance with a plan-based approach.

4.4.2 Events on the board change probabilities

The probability distribution for all unknown pieces changes if the probability distribution of one or more unknown pieces changes. Therefore the following events lead to a recalculation of the probability distributions:

- An unknown piece moves; then it cannot be a bomb or the flag
- An unknown piece is being involved with a duel and its rank becomes known.

In his thesis Vincent de Boer has described an algorithm for the recalculation of probability distributions and has given the name "normalizing of probabilities" to this algorithm [VB].

An academic research to pattern recognition in moves has shown that patterns in moves are able to give information about the ranks of unknown pieces [JS]. So pattern recognition in moves too changes probabilities and may lead to a recalculation of the probability distributions of ranks.

Human players (and probably computer programs too) often use standard concepts in their setups. If a program saves setups and games then during the progress of a game it may recognise patterns and use these patterns as an indication of the ranks of unknown pieces in an actual game.

4.4.3 Complicating factors

The recognition of bluff is an integral part of the recognition of unknown pieces from characteristics of positions or moves. There is no current theory for the recognition and handling of bluff.

Many setups and games in the Gravon database have a low level of quality. This diminishes the reliability and relevance of statistical data that come from this database. If it is possible to select setups and games on quality criteria then statistical data of that selection are more useful for the detection of real improvements than statistical data of all setups and games.

4.5 Best guess or average rank in search trees

There are two different methods of accommodating an unknown piece in a search tree:

- Average rank
  The unknown piece is processed as a weighted average of pieces with a known rank. This method had been applied in academic research of brute force search trees analysis for games with incomplete information.
- Best guess
  The unknown piece is processed as a piece with a known rank. The rank with the highest probability is chosen from the possible ranks. This method is a heuristic. It has been described by Imer Satz, the author of the program Probe that's the current world champion program.

An interesting question is whether (and if so why) this heuristic gives better results than the brute force approach. This is a subject that requires further research.

4.6 Distribution of most important ranks

Initial setups (for example setups in the Gravon database) can be classified in accordance with their distribution of the most important ranks over areas of the board. The statistical data in a classification system can be applied to infer unknown positions of other important high ranked pieces from known positions of important high ranked pieces.

4.7 Material balance

The material balance determines the long term approach of a game. For some conditions the approach is obvious:

- Decisive material gain
  Encirclement of the opponent nearly always leads to the winning of the game
- Decisive material loss
  The side with this disadvantage should stake everything and surrender if this fails
- All bombs are known
  Pieces with an invincible rank now freely can capture any piece of the opponent.
- All bombs but one are known
  If enough own pieces with an invincible rank are available then it is justified to gamble by capturing any unknown piece of the opponent with an invincible piece.

In case of a different material balance more choices in strategy are possible. No theory is available that describes which strategy is most effective under which conditions.

4.8 Control of risk

In an environment of uncertainty and lack of information actions have a speculative character. To some extent probabilities can be estimated and patterns recognized. In duels with unknown ranks the question arises what the conditions are for the taking or refusing of risk. Then not only has the calculus of probabilities to play a role. Regardless of the amount of probability serious Stratego players always will take into account what the long term perspectives are after a possible loss. Often players will abandon the capture of a piece if they can continue without risk a strategy that offers perspectives.

Both qualitative conditions and numbers are necessary for the control of risk:

- The choice with regard to an action is based on the material balance that originates if the action is performed or omitted.
- If a duel is speculative than the measure of willingness to take risks should be varied randomly, both within a game and throughout different games. This prevents that human players after playing a number of games discover regularities in the playing methods of the program and will be able to profit from these experiences.
- The valuation of the material balance depends on the scale of values that has been chosen for the valuation of pieces inclusive of their positions.

There is no theory of the limits for acceptable risk in Stratego. The use of bluff too is theoretically unexplored area. A study of gains and losses of moves to an unknown rank in the Gravon database may produce rules for what is acceptable risk and what is not an acceptable risk. Maybe methods can be used that have been proven effective in other speculative games like Poker.

4.9 A conservative or expansive approach

Two opposite strategic styles can be distinguished:

- Conservative
  Emphasis lays on control over areas, evasion of loss and engaging into save duels.
  Often a player reacts immediately on moves of the opponent in order to extract a maximum of information from events happening on the board. An important rule is to move a minimum number of pieces.
- Expansive
  Emphasis lays on the conquest of areas and pieces; if probabilities are favourable then risks are accepted. The primary goal has priority. Often outside the area of the primary goal manoeuvres or small threats of the opponent are totally ignored. Minimisation of the number of moved pieces may be rejected in favour of enlarging the chances for conquest of area or material.

Between these extremes all kinds of intermediate forms exist. Moreover it is possible to apply locally different strategies in different areas.

It's not clear which of these extremes offers the best probability to win a game. Human players are flexible and intuitively choose one of these strategies if in their opinion their choice may give best results in a certain position. Eventually they change their approach during a game.

A computer program should be able to apply both a conservative and an expansive strategy but has to make a choice, mostly locally in an area. A program cannot base its decision on intuition. The choice for conservative ⇔ expansive is being determined by strategical characteristics that have been described in the foregoing paragraphs (choice of area, invincibility of ranks, value scales for ranks, material balance and risk control).

In situations of material balance the program should vary between conservative and expansive, both within a game and throughout different games. This prevents that human players after playing a number of games discover regularities in the playing methods of the program and will be able to profit from these experiences.

Knowledge of the game is necessary to determine effective rules for the choice of a conservative or expansive approach in an area of the board. At this moment such knowledge is only available in the heads of Stratego experts. Research is necessary in order to get this knowledge available for Stratego programming.

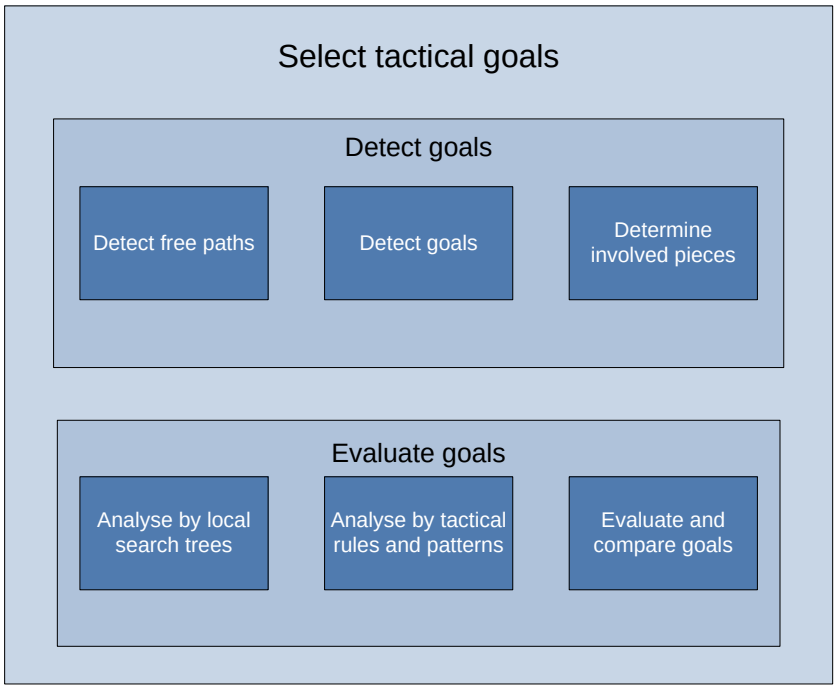
Examples of some strategical issues can be found in appendix A.

5 Middle term decision making: tactical choices

5.1 Coherence and goal

Middle term choices are directed towards coherence of moves within the framework of a strategy. Moves show middle term coherence if they are aimed at a common goal. Coherence may be present in the moves that one piece does successively. Coherence can be applied too at successive moves by different pieces. Mostly tactical goals are related to a possible duel between pieces, but also it may be of importance to have control over access to an area. The tactical issue is to detect offensive and defensive goals, to evaluate these goals and to determine priorities.

The following scheme shows the main functions that play a role in the determination of tactics.



5.2 Offensive goals

Pieces of the own side may engage into a duel with pieces of the other side. Usually a distance should be bridged before the duel has been affected. Pieces of the opponent may keep their positions, but it is possible that they will do moves in order to evade a duel.

This document contains descriptions of 15 offensive themes in appendix B:

- Goals in the initial phase
- Explore significant spots
- Explore significant pieces
- No sideway mobility
- Restricted sideway mobility
- Encirclement
- Bulldozer attack
- Chase with a fork
- Explore and conquer
- Control in an area
- A siege
- Exchange of a defender
- Bind a high rank
- Stalking at the right moment
- Forced moves.

Most of the offensive themes are about duels and in most themes a restricted number of pieces are being involved directly in an eventually possible duel. Involvement of pieces may extend to pieces that are not directly engaged into a duel. Examples:

- It may be necessary to mobilise a piece by moves of a blocking piece
- It may be favourable to have more freedom of choice of ranks for a duel and therefore to advance extra pieces towards the target square of a duel.

Besides the conquest of pieces the conquest of positions too may be of importance. Mostly this relates to a position for a piece with a high rank. In that position the high ranked piece realises a free passage for pieces with lower ranks.

The choice of offensive goals too is being determined by the current strategy. An expansive strategy is directed to the conquest of terrain. In that case offensive goals lay in the same area. In case of a conservative strategy the choice of goals is not restricted to an area and priority is given to the save discovery and conquest of high ranked pieces.

A material unbalance (both gain and loss) may be a reason to accept more risk in the capture of unknown pieces. Preferably such high risk attacks are directed to squares with the lowest probability of an unfortunate outcome. For human players it is difficult to maintain a matrix of probabilities for ranks, a program can do this job much better by using statistical data from the Gravon database and maintain probabilities with the normalization algorithm by Vincent de Boer.

Bluffs may be done in offensive actions but in most offensive themes the results of bluffs disappear as soon as a bluff has been unmasked. Only in a few offensive themes a bluff offers significant opportunities for a material gain (An example is the bulldozer attack). The power of bluffs mostly lies in a temporary disturbance of actions by the opponent and there the bluff primarily serves a defensive goal.

In his thesis Vincent de Boer describes a few offensive plans [VB] and mentions that more plans are necessary in a Stratego program.

Probably the list of offensive themes is not complete. If the reader of this document knows more offensive themes then suggestions are welcome.

5.3 Defensive goals

Pieces of the own side can be the target of pieces of the other side, specifically if they have been moved or if their rank is known. Such targets are still more suitable to attack if the opponent possesses (locally) invincible pieces.

Involvement of pieces may extend to pieces that are not directly engaged into a duel. Examples:

- A high ranked piece can block access to a threatened piece thereby preventing that a duel will take place
- A piece with a low rank can be placed before a high ranked piece in order to prevent the detection of the rank of the high ranked piece.

Various methods are available to prevent the loss of a piece:

- Evasion of a threat
- Protection by a high ranked piece
- Interception of the attacker
- Mobilize potential targets
- Long distance protection by a high ranked piece
- Explore the unknown rank of a candidate attacker
- Prevent detection of a high ranked piece.

Bluffs may be used to simulate protection.

Sometimes a complete reorganisation is necessary to realise an effective defence. Particularly this occurs when own moved or known pieces may be lost against a piece with an invincible rank.

The list of defensive themes probably is not complete. If a reader of this document knows relevant extensions then suggestions are welcome.

Appendix C shows examples of defensive goals.

5.4 Rule based decision or local goal directed search trees

5.4.1 The choice of a decision method

On the tactical level game knowledge is complex and voluminous. By lack of game knowledge a program author often has to start from nothing. An elementary and probably incomplete set of rules may be used as the starting point for a long period of trial and error. The progress in understanding and knowledge of the game requires a corresponding stream of changes in the decision structure. Serious issues are the survey-ability and maintainability of data in the decision structure.
There are methods to record game knowledge in decision structures:

- Decision rules can be hard coded in a program
  This is can be done for rules that are resistant to changes by progress in game knowledge
- Decision tables that relate conditions to actions
  Values in such structures can be changed and extended, but as the volume grows the burden of their maintenance may grow exponentially
- Expert system
  Such systems normally offer means to manage the burden of maintenance for large volumes of rules. A description of the implementation of an adaptable expert system in Stratego can be found in the thesis of Mohannad Ismail [MI].

Complexity, volume and progress by trial and error suggest the use of an expert system for tactical rules and decisions.

But it is rather simpler to use local goal oriented search trees:

- Determine which pieces of the own side are involved with the duel
- Determine which pieces of the other side are involved with the duel
- Evaluate a local search tree of goal directed moves with the pieces that are involved.

The concept of local search trees is not new; for example under various names like “tactical search” it has already been applied in some Go programs [WP2]. This concept perfectly suits to Stratego because many tactical actions in Stratego are restricted to a limited number of pieces and a delimited area of the board.

Probably it is not possible to cover all aspects of tactical decision making by local search trees. A hybrid of partly local search trees and partly game rules probably offers here the optimal solution.

For more information refer to:

- local search trees [WP3]
- expert systems [WP4]
- decision tables [WP5].

5.4.2 Selective or brute force

It is possible to include all legal moves in the search tree or to apply selection of moves. Selectivity may be applied to pieces, but also it is possible to select on move characteristics, for example only forward and sideward moves.

Selectivity may lead to the overlook of tactical opportunities. But selectivity too may lead to the detection of more deeply hidden opportunities by enlarging the search horizon. No literature about the effects of selectivity in search trees of Stratego is available, so considerations about this subject are speculative. Maybe selectivity works better in Stratego than in games with complete information like chess. Because at this moment there is no certainty about the effects of selectivity this is an interesting area for research.

5.4.3 Improving efficiency in goal oriented search trees

An issue is that the distance between goal related pieces may be too large. Then the local search tree mostly consists of moves that bridge a distance. This may require so many moves that relevant duels or positions will not lie within the horizon of the search tree. Possible solutions for restrictions of the horizon are:

- Use “jump moves” instead of normal moves in search trees
- Use selectivity in order to increase the search depth
- Use static look ahead functions to prune the search tree.

These ways to improve efficiency in a local tree search are illustrated in appendix D.

5.5 Algorithms that provide static look ahead

Probably it is possible to apply static look ahead exactly in the following situations:

- More than two pieces engage into a duel.
- An attacker threatens a piece with a restriction to sideway moves.
- A piece can be encircled.
- A bulldozer attack.
- Attack with a fork.
- Interception of an attacker that tries to attain a target.

If this kind of situation is present in a game the result of a tactical action is obvious for a somewhat advanced Stratego player. Then it should be possible to define formal rules for these situations and to program code from these rules. Probably the development of algorithms for these situations will result into a relevant improvement of the playing strength.

In appendices B and C some examples can be found of possible types of look ahead.

5.6 Evaluation of goals

The evaluation of goals (by static look ahead of a local search tree) leads to values for the goals. The values can be compared and so the priority of goals can be determined. Moves that are required for a goal get a value that is deduced from the value of the goal.
Literature does not mention anything about the valuation and evaluation of goals so knowledge about this aspect of the game has to be developed from nothing.

6 Short term decision making: the final choice of a move

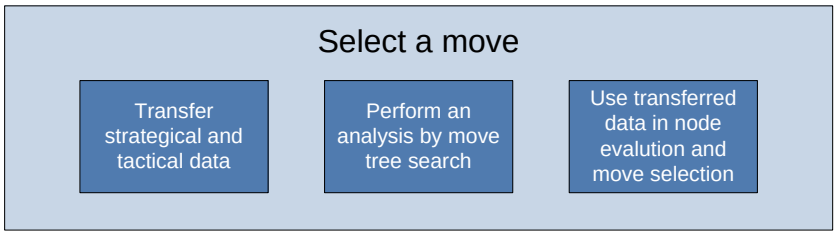
6.1 Search trees without top-down strategy and tactics

This is the method that has been implemented in academic research of search trees in Stratego. No or almost no attention has been spent to separate functions for decision making on the level of strategics and tactics. In the search tree moves are evaluated by material or positional factors. All possible moves are included in the tree; principally the analysis of the tree is brute force. In most studies there is a lack of functions that expand the horizon by static look ahead.

6.2 Search trees with top-down strategy and tactics

The strategical and tactical analysis produces best moves from local tree searches or from rule based decision methods. A choice out of these best moves should be made. Other kinds of data too can be made available. These products of the pre-analysis should be used in the final tree search that determines the best move.

The following scheme shows the main functions that play a role in the move selection.



6.2.1 The transfer of best moves

The pre-analysis by local search trees or rule based decision methods produces corresponding best moves. These moves may be used as selection criteria in the root position of the final tree search.

6.2.2 The transfer of other moves

It is possible to restrict the final search tree to moves that have been present in one or more of the local tree searches.

6.2.3 The transfer of node values

Transposition tables may be used to transfer node values from local search trees to corresponding nodes in the final tree search. For a short discussion of transposition tables refer to [SA].

6.2.4 Positional values

It is possible to assign rank or even piece specific positional values to squares on the board. This may be used as a method to stimulate moves to squares that lead to goals over the horizon of a move tree search.

6.2.5 Brute force or move selection (or both)

The final tree search analysis can be realised as a brute force analysis or as an analysis with some kind of move selection. It too is possible to apply a search with selection first and repeat the same analysis without selection as long as processing time allows.

6.2.6 Predictability

If the program has the choice between nearly equivalent moves than the program should make a random choice. This prevents the detection of rigid habits and the use that can be made of them.

7 Overview of opportunities for improvement by game knowledge

Here follows an overview of:

- starting points for improvement in Stratego by game knowledge
- the status quo of academic literature and / or research
- an estimate of the impact of improvements with regard to the status quo.

Starting point for improvement and strengthening	Literature and research	Impact
Classification of initial setups	None	Small
Generating and valuation of initial setups	Some incomplete information	Small
Valuation of ranks	Some incomplete information	Middle
Theory of probability for unknown ranks	Has been worked out	Small
Recognition of ranks with patterns in game positions	Only bomb flag patterns	Large
Recognition of ranks with patterns in moves	Some incomplete information	Large
Handling of unknown ranks in search trees	Some incomplete information	Large
Material balance	None	Middle
Control of risk	None	Large
Conservative or expansive approach	None	Middle
Choice of goals and pieces involved with goals	None	Large
Local search trees	Only in other games	Large
Static look ahead	Some incomplete information	Very large
Selectivity	None	Large
Jump moves	None	Large

Some relevant categories are not mentioned explicitly in this scheme:

- Bluff
- The use of game knowledge by means of an expert system.

They are not mentioned apart because they are an integral part of the points of improvement.

Bluff can be worked out in three points of improvement:

- Recognition of ranks with patterns in game positions
- Recognition of ranks with patterns in moves
- Control of risk.

The use of game knowledge by means of a tailor made expert system can be worked out in the following points of improvement:

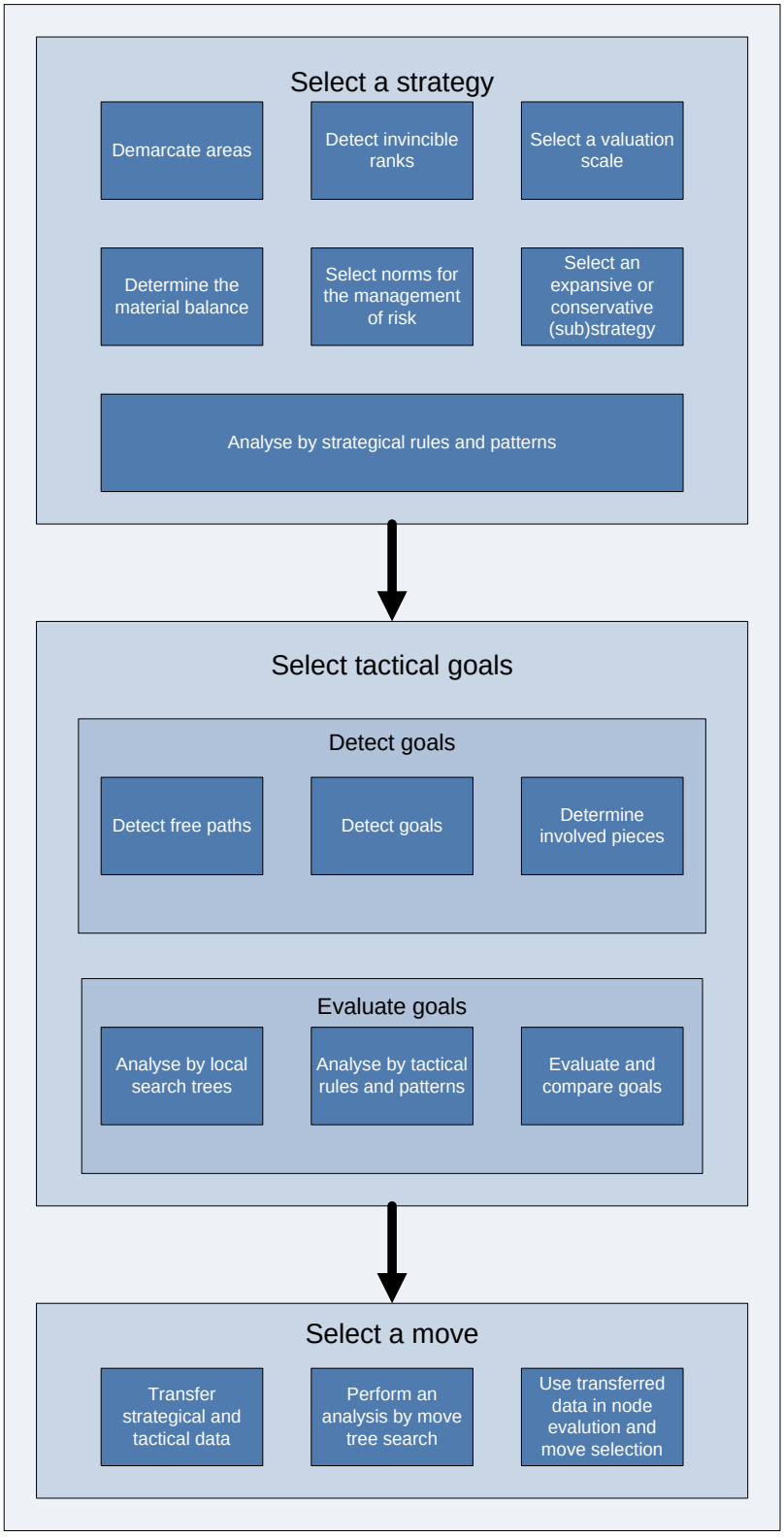
- Valuation of ranks
  Determination of the right valuation scale for ranks
- Recognition of ranks with patterns in initial game positions
- Recognition of ranks with patterns in moves
- Determination of the material balance
  The qualitative part that cannot be determined by numbers and computation
- Control of risk
  The qualitative part that cannot be determined by numbers and computation
- Choice between a conservative or expansive approach
- Determination of goals and pieces involved with goals.

Probably the data representations of game knowledge are specific for these subjects. In that case corresponding specific and apart expert systems will be required.

8 Concept of a Stratego program with effective improvements

8.1 The authors opinion

A top-down approach from strategy to tactics and the choice of a best move in a search tree can be realised in a Stratego program with the following processes:



The use of game knowledge is crucial for improvement of the playing strength. Strategical decisions are rule based. Tactical decisions are both rule based and tree search based. The final choice of a move is search tree based. So game knowledge only is required for strategic and tactical decisions. More specifically a tailor made expert system will be required for the following processes:

- Determination of the right valuation scale for ranks
- Determination of the material balance
- Management of risk
- Choice between a conservative or expansive approach
- Determination of tactical goals and the pieces involved
- Recognition of patterns in initial game positions
- Recognition of patterns in moves.

Additionally it will be necessary to define a classification method for initial setups and to develop a function that generates initial setups for a program that works as mentioned above. Besides that it may be profitable to record the results of games with opponents and use them in future games.

8.2 Motivation of choices

The most important reason for the (partially) implementation of search trees is that this method takes away a part of the complexity of the decision process. The use of a best guess for the ranks instead of an average of ranks looks attractive because it diminishes complexity, it extends the horizon of search trees and besides that it will fit to an eventual Monte Carlo approach. But it is an approach that requires research and theoretical consideration. Selectivity of moves in Stratego is another subject that may be of importance from a theoretical point of view.

If you wish to make a program with only look-ahead then this requires a complex and voluminous decision structure that is difficult to create and maintain. For the author of this discussion such an approach is unattainable.

The same argument applies to the method of using agents. This requires an extensive implementation of game knowledge by rules for each agent. Maybe this burden can be reduced by the use of an agent “headquarter” that handles overall strategy and tactics and communicates tasks to the agents that represent pieces.

The author has less reserve with regard to the Monte Carlo method. But the author assumes that an implementation of this method should be built on experience with a search tree based approach. Maybe after the fulfilment of a tree based approach the author will extend activities to a Monte Carlo approach. A specific variation – Monte Carlo tree search – has been implemented successfully in various GO programs. The value for Stratego is unknown. Probably this domain-independent method will be the subject of an academic research project sometime, maybe it already is now.

Genetic algorithms may be used successfully to round off the optimisation of evaluation functions. The use of game knowledge to define relevant criteria should precede the optimising process for a set of criteria. The author assumes that statistical analysis of the Gravon database is the best way to detect winning criteria for moves and initial setups.

So the author has a definite preference for a hybrid of tree search and rule based decision.
The availability of more literature about tree search in Stratego than other approaches has been an extra stimulus for that choice.

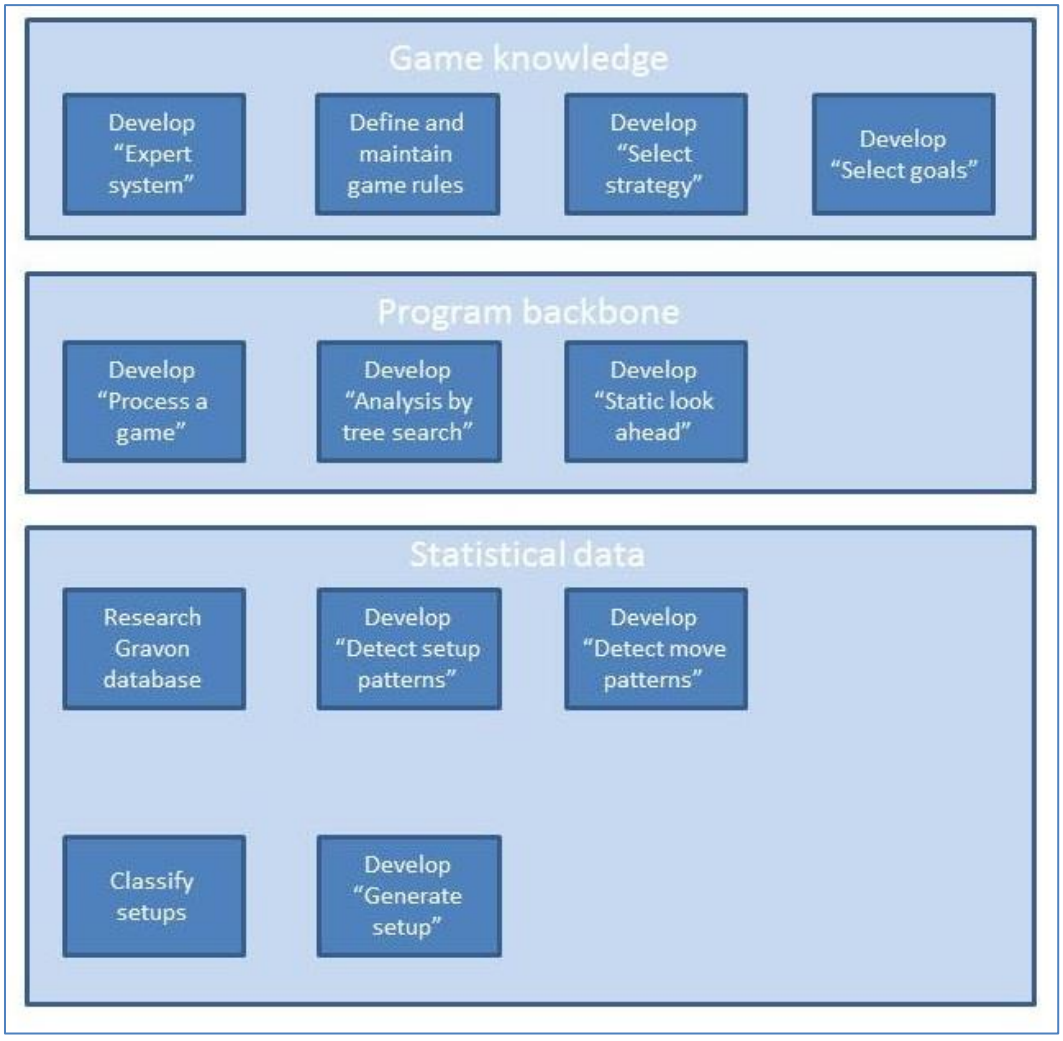
8.3 Realisation

Here follows an overview of estimations of investments to be done for points of improvement.

Starting point for improvement and strengthening	Improvement	Investment
Classification and valuation of initial setups	Small	Large
Generating of initial setups	Small	Large
Valuation of ranks	Middle	Middle
Theory of probability for unknown ranks	Small	Small
Recognition of ranks with patterns in game positions	Large	Middle
Recognition of ranks with patterns in moves	Large	Large
Handling of unknown ranks in search trees	Large	Small
Material balance	Middle	Large
Control of risk	Large	Large
Conservative or expansive approach	Middle	Small
Choice of goals and pieces involved with goals	Large	Small
Local search trees	Large	Middle
Static look ahead	Very large	Large

Large means more than 1 year, middle is about 6 months, small is about 2 months.

This will require global activities as shown in the next scheme.



At this moment many of the activities in this scheme have not been done nor are descriptions available that give adequate information for the building of a program. Most of the improvements require some kind of research. Research is necessary for real knowledge of what is effective and best practise. The Gravon database is a precious source of statistical data that may be applied to most of the points of improvement.

Some progress already has been made by the author:

- Classification of initial setups
- Generating and valuation of initial setups
- An algorithm that handles the two-squares rule (Part of static look ahead)
- Valuation of ranks.

At this moment a study of the material balance is being done. A next subject will be the choice of goals and pieces involved with goals. This can be combined with the development of algorithms for additional kinds of static look ahead.

The requirements in this document show that the making of a Stratego program is a large project. The project activities require a considerable amount of time, most of it being preliminary effort. For the author of this discussion it will take many years to accomplish all activities in this scheme. The building of a Stratego program still is far away. But why hurry? Omission of one (or more than one) of these activities will diminish the chances to make a program that plays on a reasonable level. If time, health and motivation allow it someday the job will be done.

8.4 Legal aspects

Jumbo owns the rights of the game concept and artwork of Stratego. It is forbidden to use the artwork of Jumbo Stratego in programs or documents or to attach the name Stratego to a program without the explicit permission of Jumbo.

Some Stratego programs with own artwork and names exist and are available on Internet and it is usual to mention that a program plays Stratego. This shows that in practice no restrictions are in force now (and maybe in the future too) to the publication of non-commercial programs that play Stratego.

8.4.1 Acknowledgements

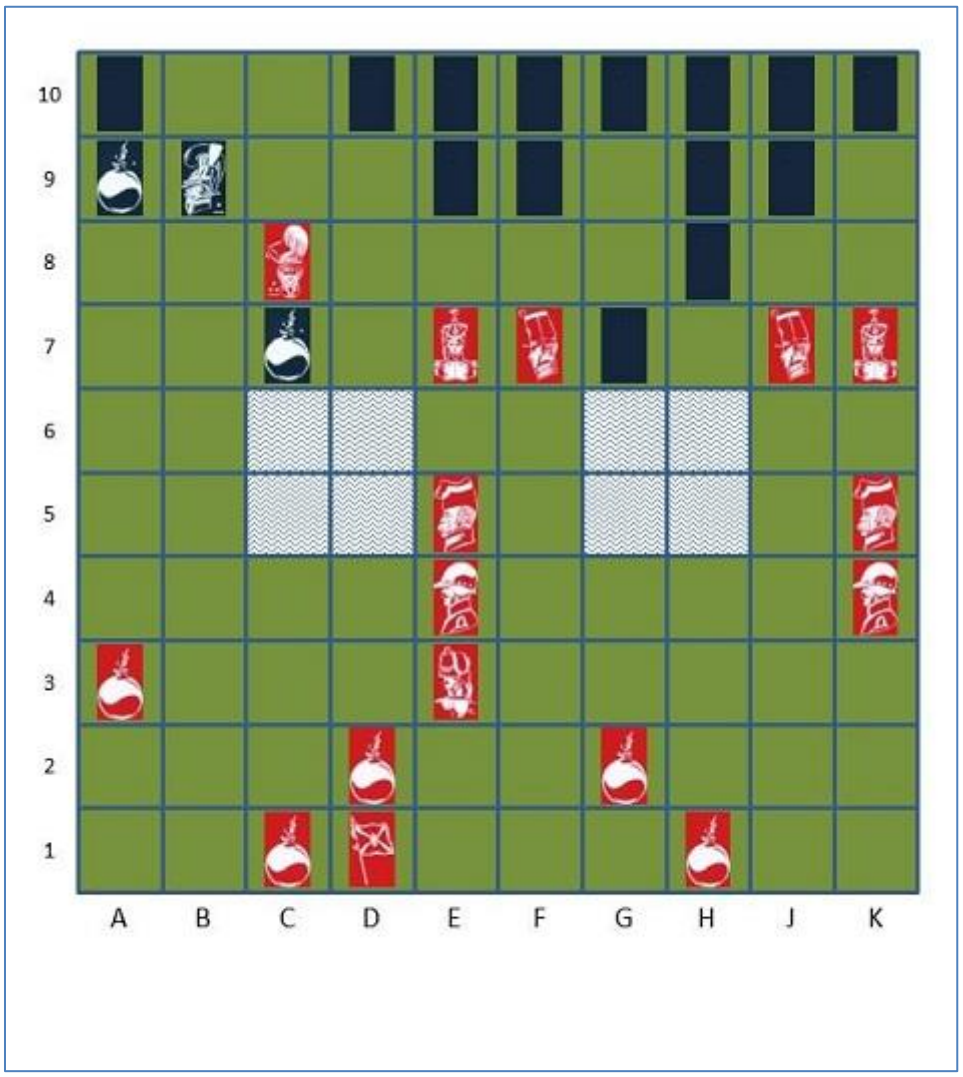
The author appreciates that by courtesy of Jumbo it is possible to use pictures with the original artwork of Jumbo Stratego in this document.

9 Literature references

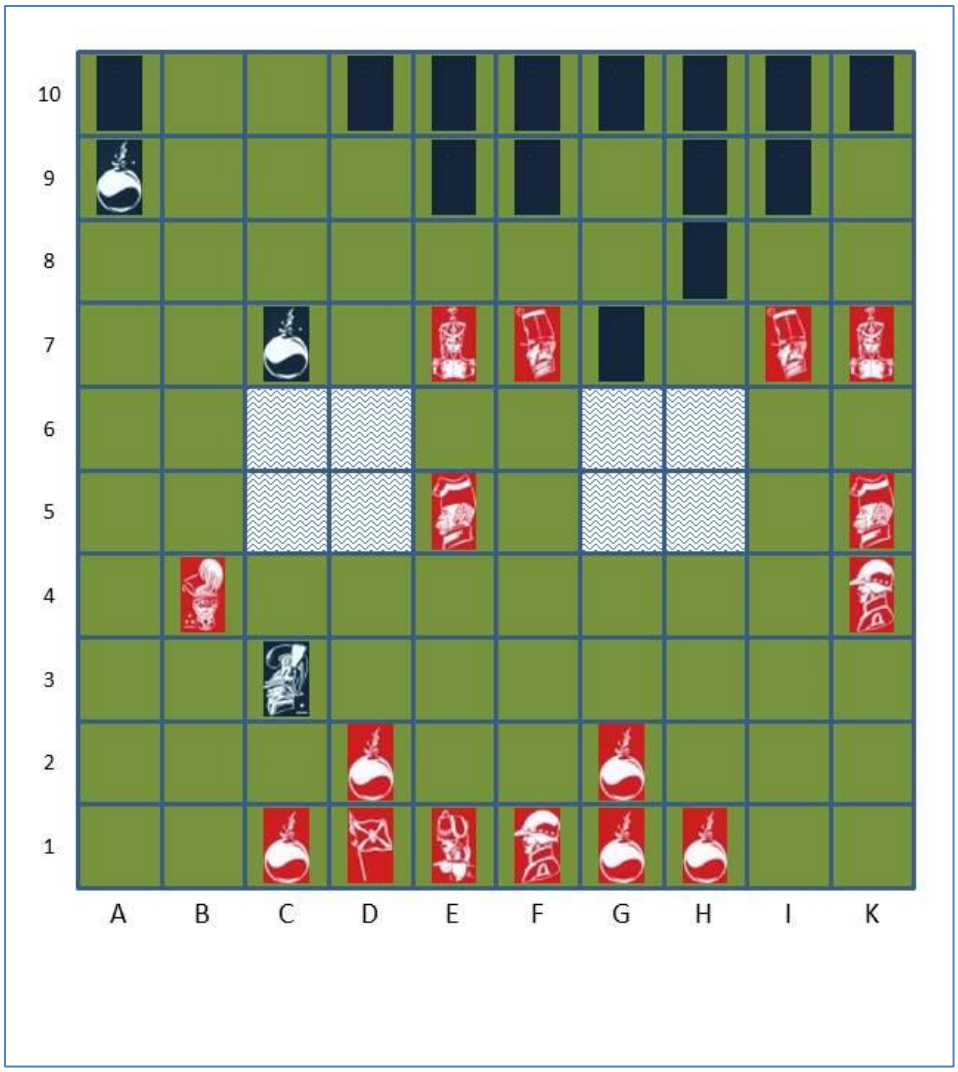
Reference	Description
AP1	Aske Plaat, Research, Re:Search en Re-search 1996
IS	www.probe.imersatz.com/tutorials/Probe_Tutorials.swf (Does not exist anymore)
JM1	Jeroen Mets, Monte Carlo Stratego, 2008
JM2	J. Mohnen, Using Domain-Dependent Knowledge in Stratego, 2009
JS	Jan A. Stankiewicz en Maarten P.E. Schadd, Opponent Modeling in Stratego, 2009
KS	Karl Stengård, Utveckling av minimax-baserad agent för strategispelet Stratego, 2006
MI	Mohannad Ismail, Multi-agent Stratego, 2004
MS	Maarten P.E. Schadd en Mark H.M. Winands, Quiescence Search for Stratego, 2009
PT	Jeff Petkun en Ju Tan, Senior Design Project Stratego, 2009
RA	Ryan Albarelli, Optimizing Stratego heuristics with genetic algorithms, 2003
RB	R.M. de Boer, Reachable level of Stratego using genetic algorithms, 2012
SA	Sander Arts, Competitive play in Stratego, 2010
SS	Schuyler Smith, Learning to play Stratego with convolutional neural networks, 2015
VB	Vincent de Boer: Invincible, A Stratego bot, 2007
VT	Vincent Tunru, Feasibility of applying a genetic algorithm to playing Stratego, 2012
WP1	http://en.wikipedia.org/wiki/Search_algorithm
WP2	http://en.wikipedia.org/wiki/Computer_Go
WP3	http://en.wikipedia.org/wiki/Local_search_(optimization)
WP4	http://en.wikipedia.org/wiki/Expert_system
WP5	http://en.wikipedia.org/wiki/Decision_table
WP6	http://en.wikipedia.org/wiki/Shortest_path_problem

Appendix A: Some examples of strategical issues

A.1 A large material preponderance

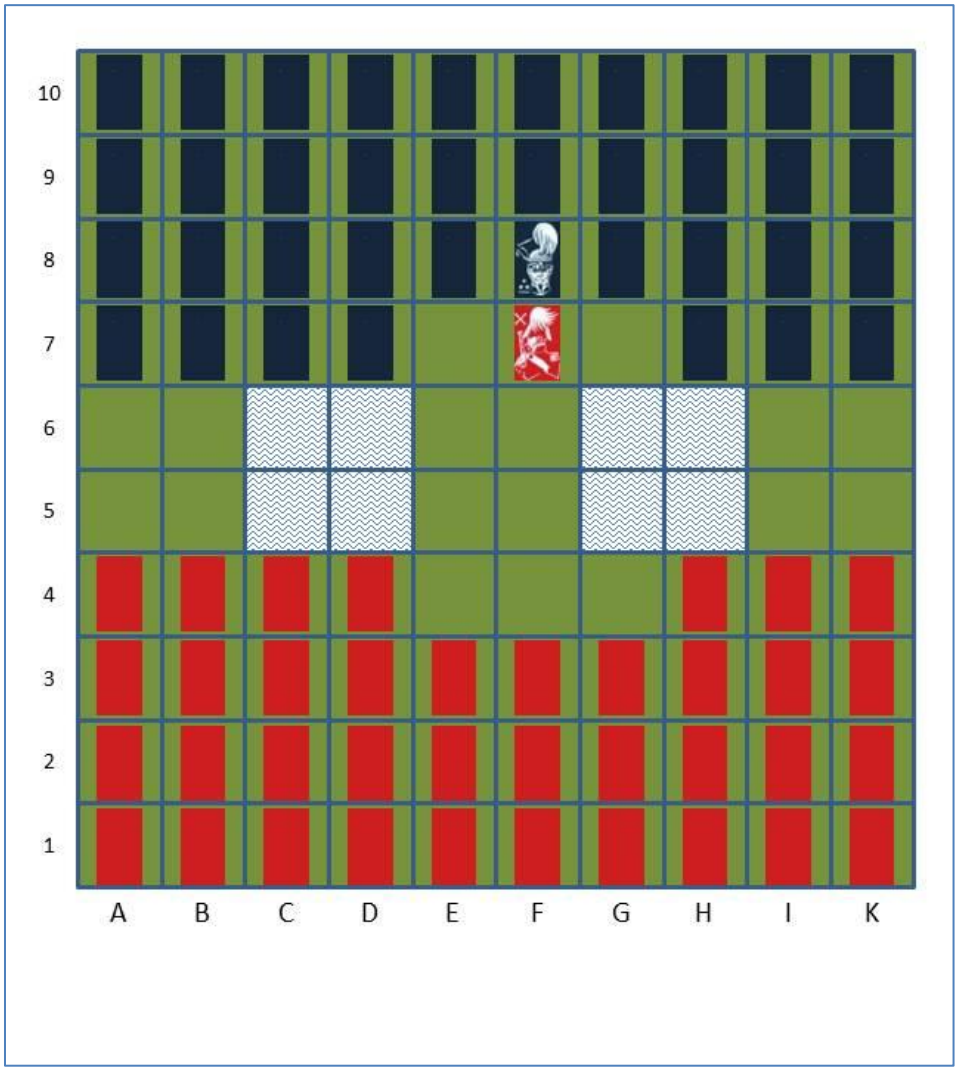


The preponderance of red consists of 1 colonel + 2 captains against 1 blue major.
Red controls the lanes. By this encirclement blue is not able to evade the final loss of this game.



In this game position too red has a preponderance of 1 colonel and 2 captains against 1 blue major. But here the encirclement is not complete. Blue should undertake a kamikaze action with the major and find the red flag or lose the major on a bomb.

A.2 To capture or not to capture, that's the question



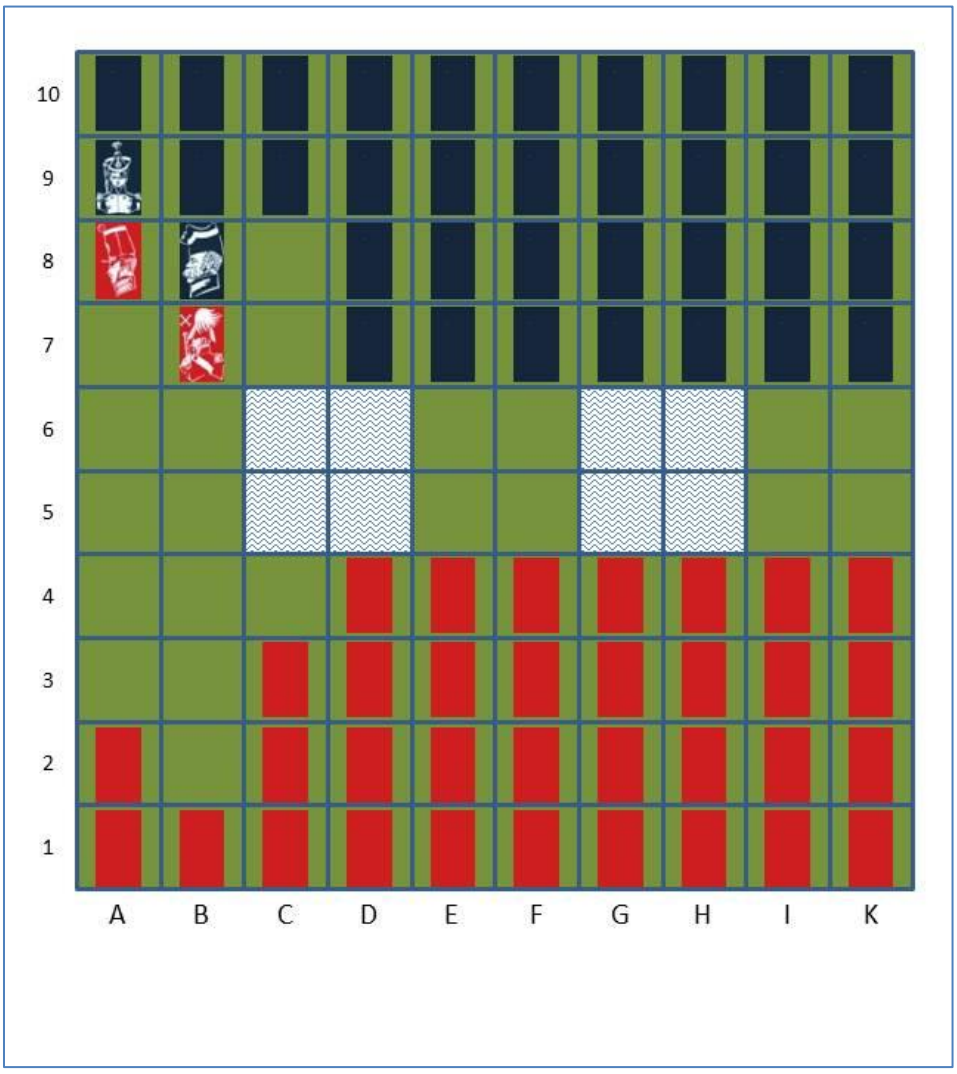
There is a material balance and the rank of the blue colonel is known. Take or leave?

There is no need for an immediate capture. So red should postpone the conquest of the blue colonel. If an escape of the colonel or exchange of marshals becomes possible then again red should make a decision about this choice.

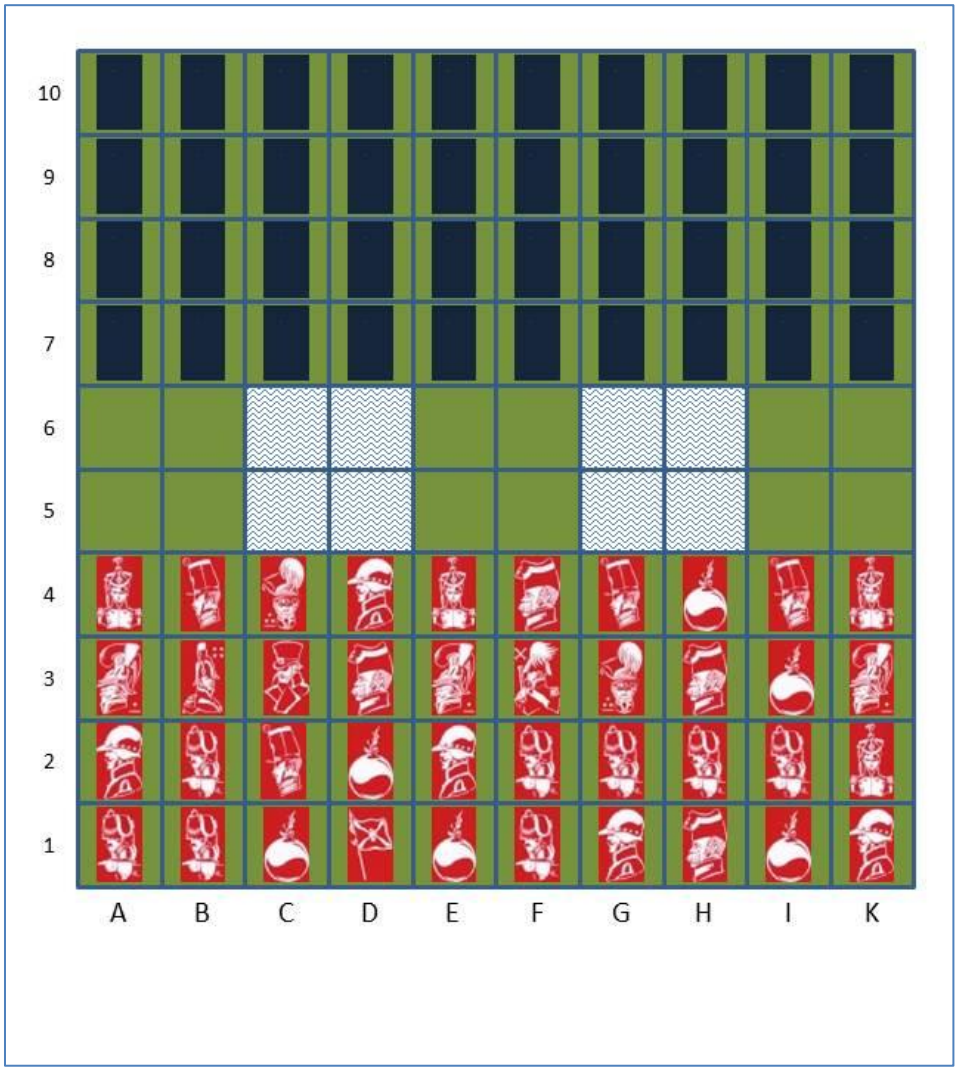
It is important to detect the position of the blue general. If red takes the blue colonel and loses the marshal then generals should be exchanged as soon as possible.
This would lead to a position with 2 red colonels + a red spy against 1 blue colonel + 1 blue marshal. Strategically this is a position with equal chances for both sides.

A.3 Strategically preference for a capture

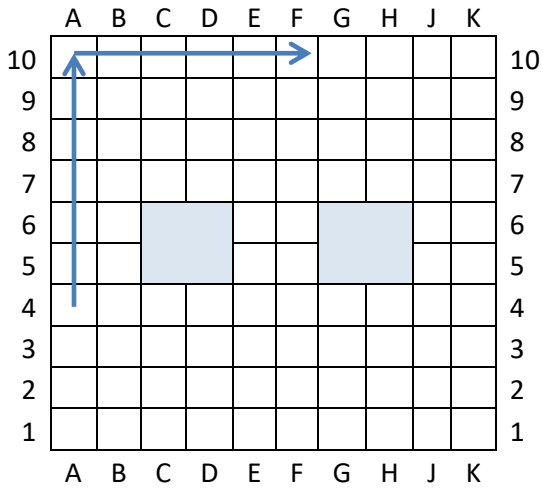
The red captain may capture the blue sergeant or the blue lieutenant. The conquest of the blue lieutenant may lead to the loss of the captain. This will slow down the conquest of terrain in the left area of the board. In an expansive strategy the efficient and quick conquest of terrain is more important than the material gain of a lieutenant. Red should take the blue sergeant and capture the unknown piece on B9. This manoeuvre will lead to the save conquest of square B8 for the marshal. On that position the marshal protects the march of lower ranks over the A line. This enables a march of red over the tenth row. Blue can only resist this kind of march if blue has chosen an initial setup that was specifically meant to resist a march through the back row.

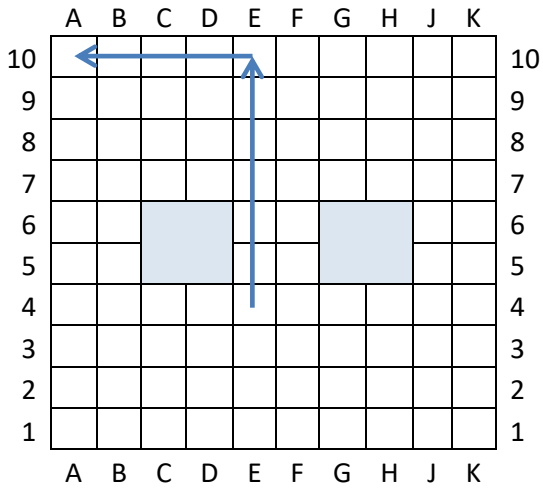


A.4 An initial setup that enables an expansive strategy



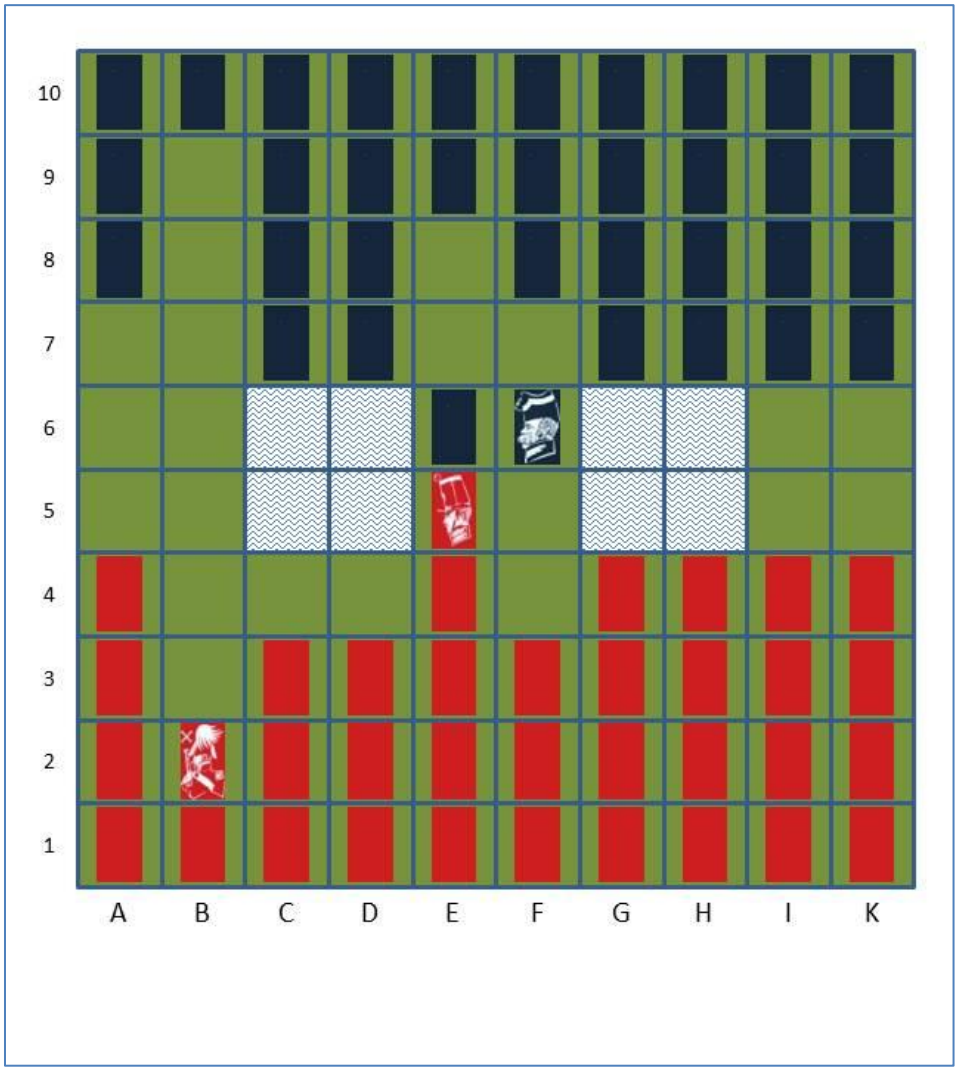
A locally expansive strategy is possible in both the left and the middle lane. Possible march routes:





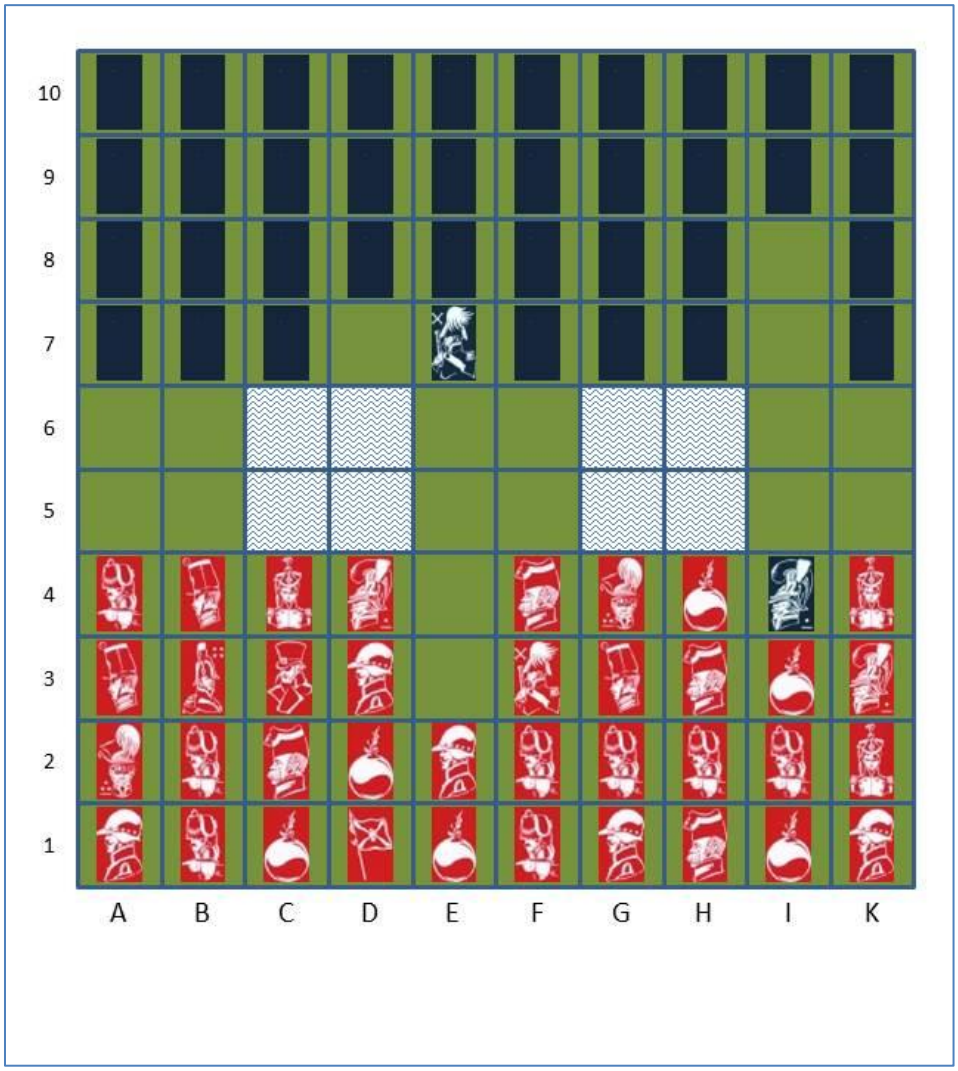
Small adaptions to these routes are possible in order to diminish predictability.

A.5 Rank of the own marshal is known



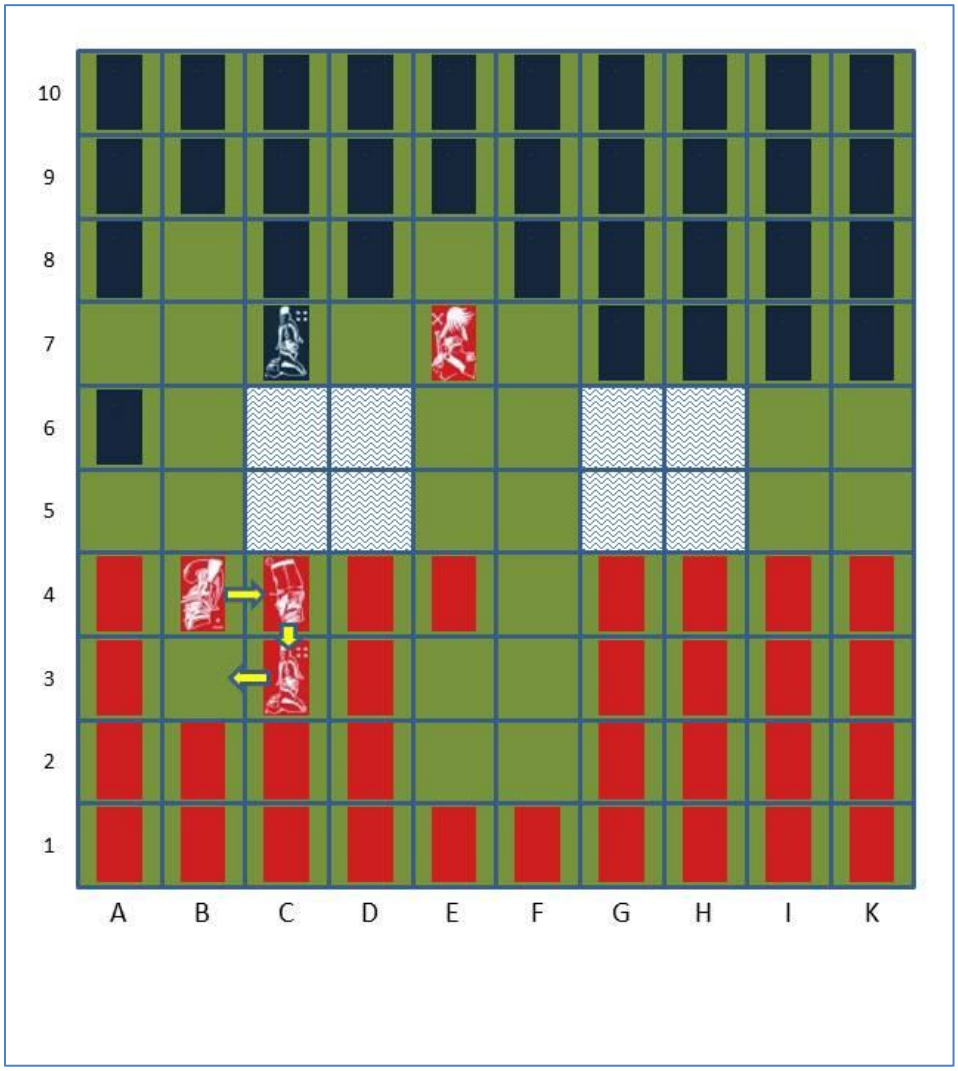
Until now red has kept to a conservative strategy. A blue scout just has discovered the rank of the red marshal. Now red should change to an expansive local strategy in the left area.

A.6 Rank of the opponent marshal is known



The rank of the blue marshal just has been discovered.
Red should change the locally passive strategy in the left area to an expansive local strategy, because the general will be locally invincible as long as blue pieces are blocking access to the red general.

A.7 Defence against an invincible general

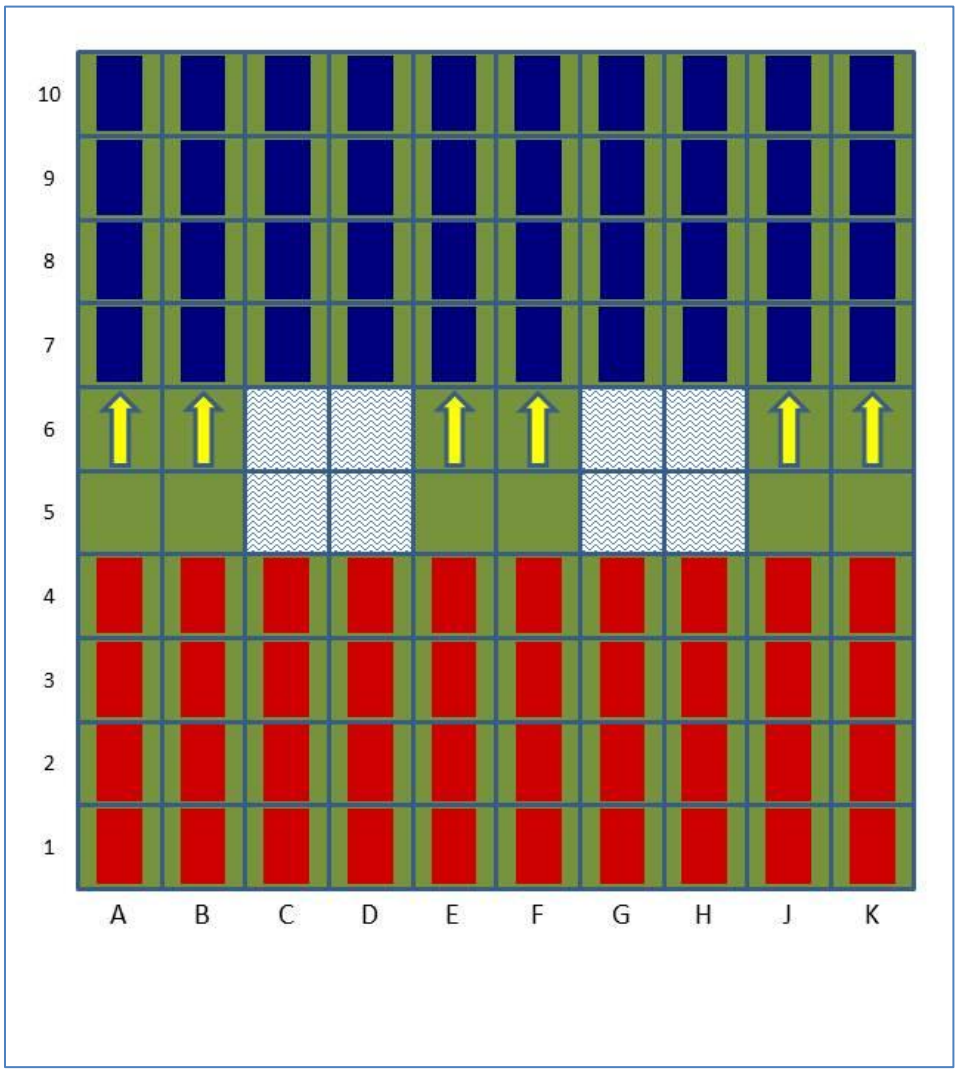


As soon as a piece of the opponent has gotten invincibility defensive measures should get priority. Sometimes extensive measures are necessary.
In this example the ranks of the blue general and red marshal just have been discovered. The rank of the red major on B4 is known. Only a direct and complete reorganisation of piece positions prevents the loss of the red major in this game position.

Appendix B: Some examples of offensive goals

B.1 Discover ranks

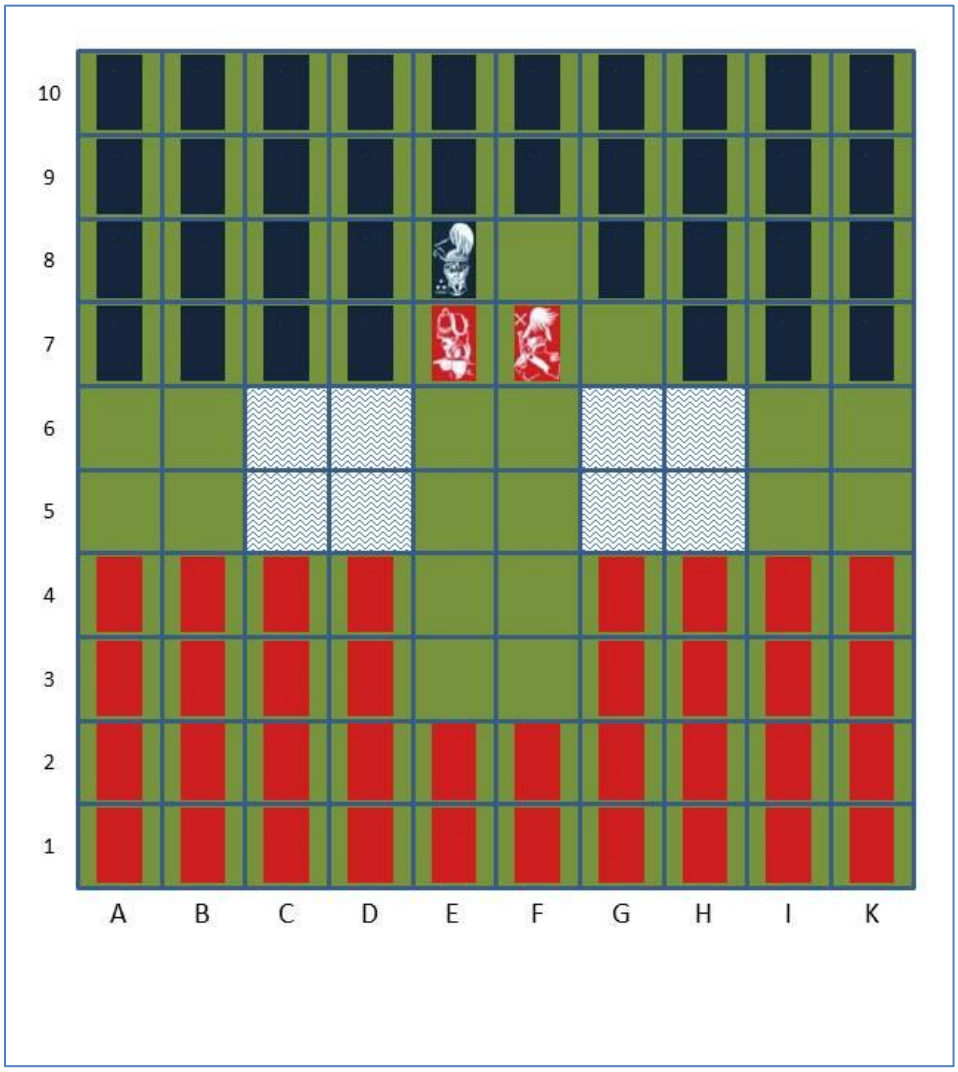
B.1.1 Initial setups



The arrows show potential targets in the initial position.
No literature is available about this situation. The following is the author's opinion. It is possible that real Stratego experts have other ideas.
The front positions mostly are occupied by sergeants, lieutenants or captains. Other ranks are less effective. The ranks of potential attackers too should be sergeant, lieutenant or captain.
Other ranks should not be used to explore unknown pieces:

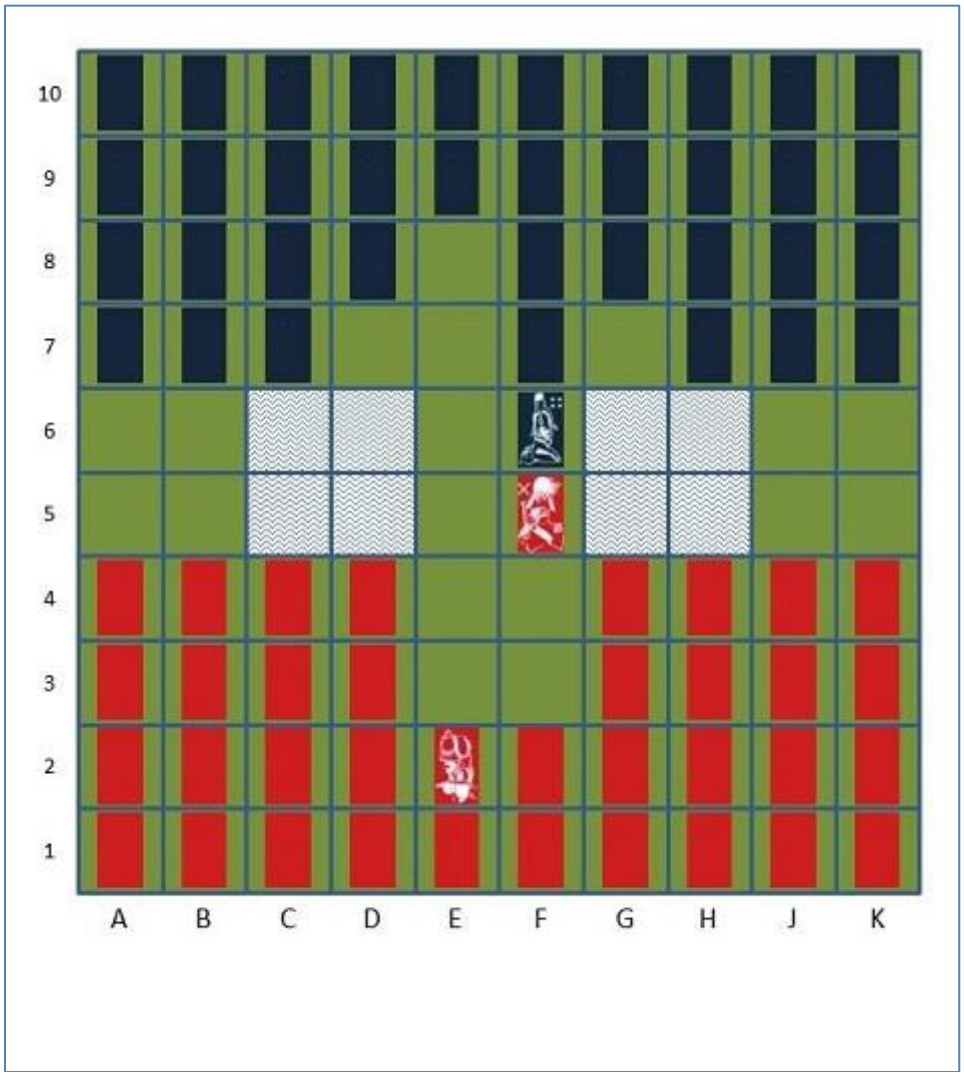
- Scouts
  Their long distance potential is wasted and the revenues are low.
- Miners
  They are needed for the removal of blocking bombs
- Otherwise
  Too much value is at stake. The loss of a major is compensated only by the discovery of a marshal or general of the opponent. If a major is lost to a colonel then this only is acceptable if the colonel of the opponent will be captured.

B.1.2 Explore an eventually relevant spot



If the rank of a blue colonel on E8 becomes known then probably the rank of an unknown piece on D7 is interesting. There is a small but definite probability on the pattern of a triangle consisting of a spy on D8, a general on D7 and a colonel on E8.

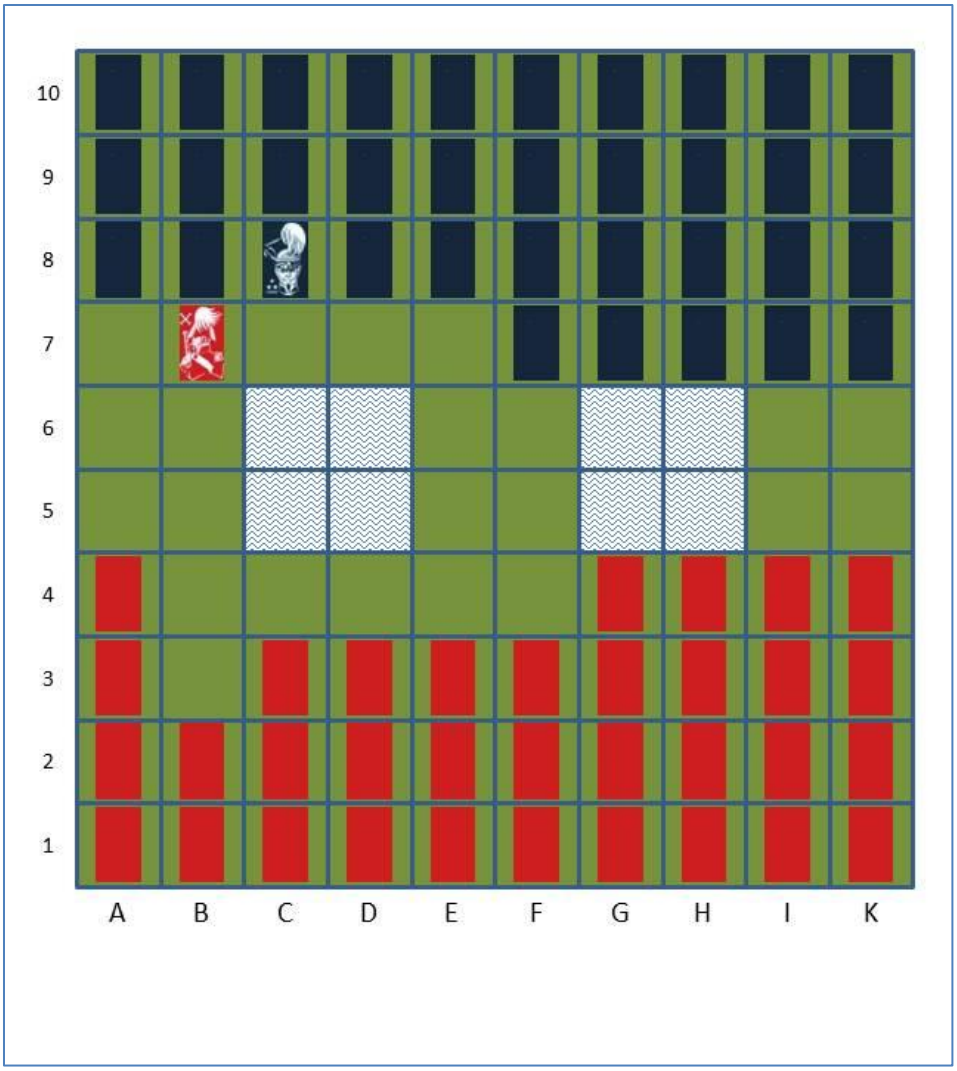
B.1.3 A search for the spy



The scout on E2 moves to E7 in order to check what rank stands on F7.

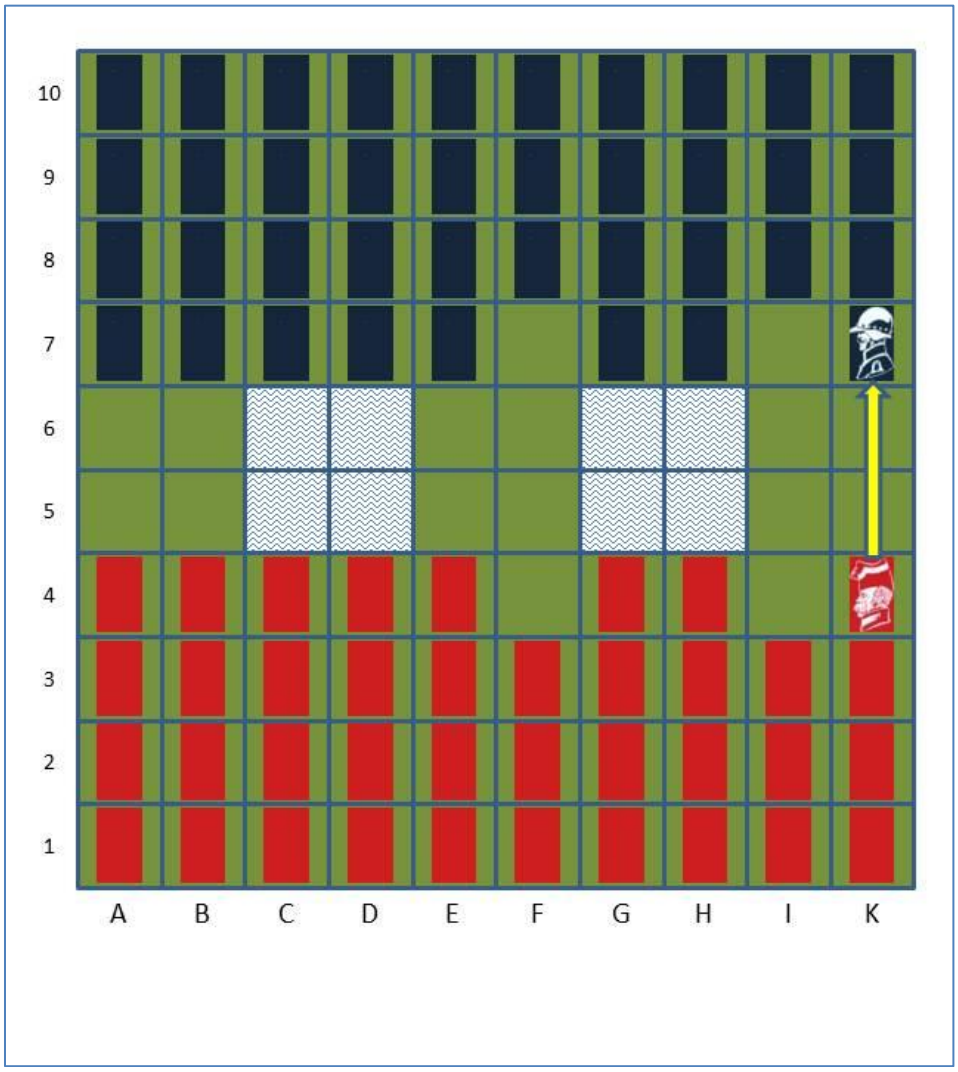
B.2 Conquer material

B.2.1 No sideway mobility at all

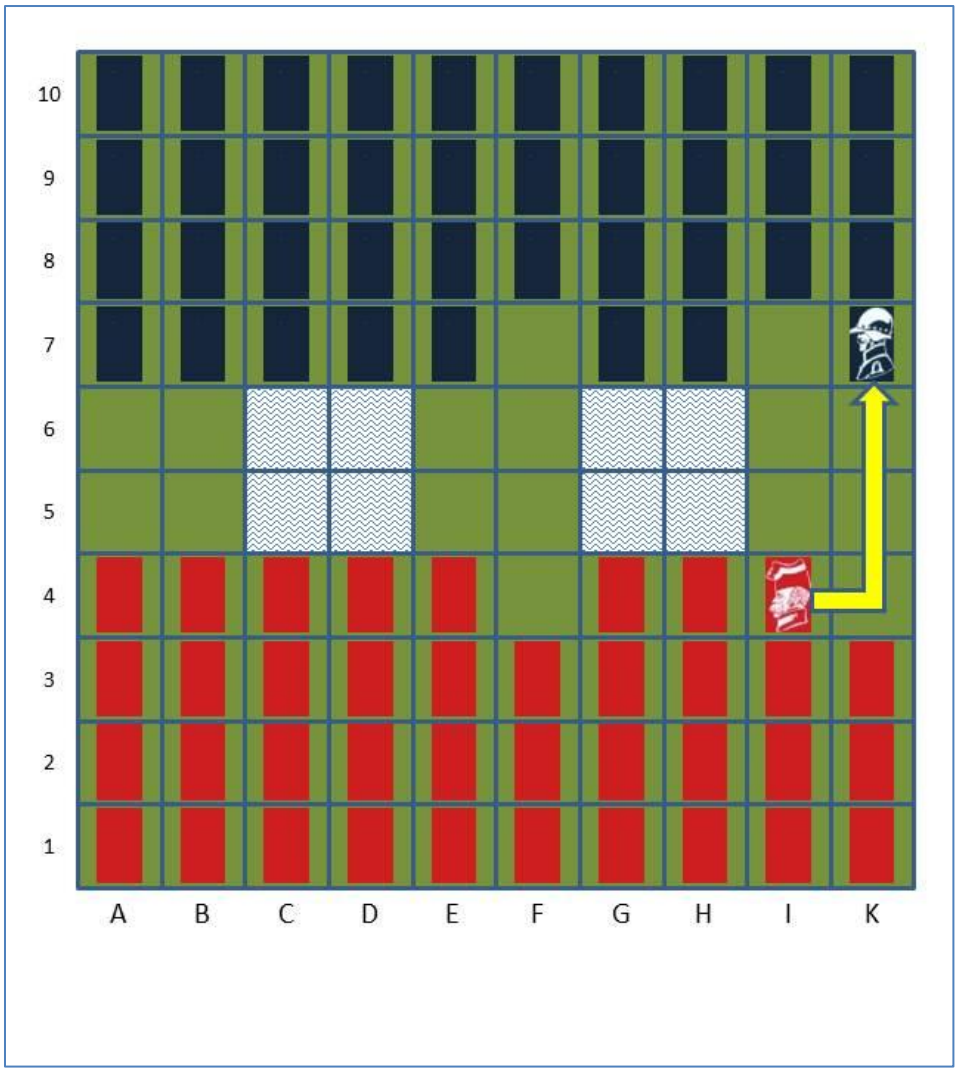


The marshal moves to C7. The colonel on C8 cannot make a sideway move and can be captured. This kind of capture can be foreseen and is suitable for the implementation in a static look ahead algorithm.

B.2.2 Sideway mobility restricted to two columns or rows

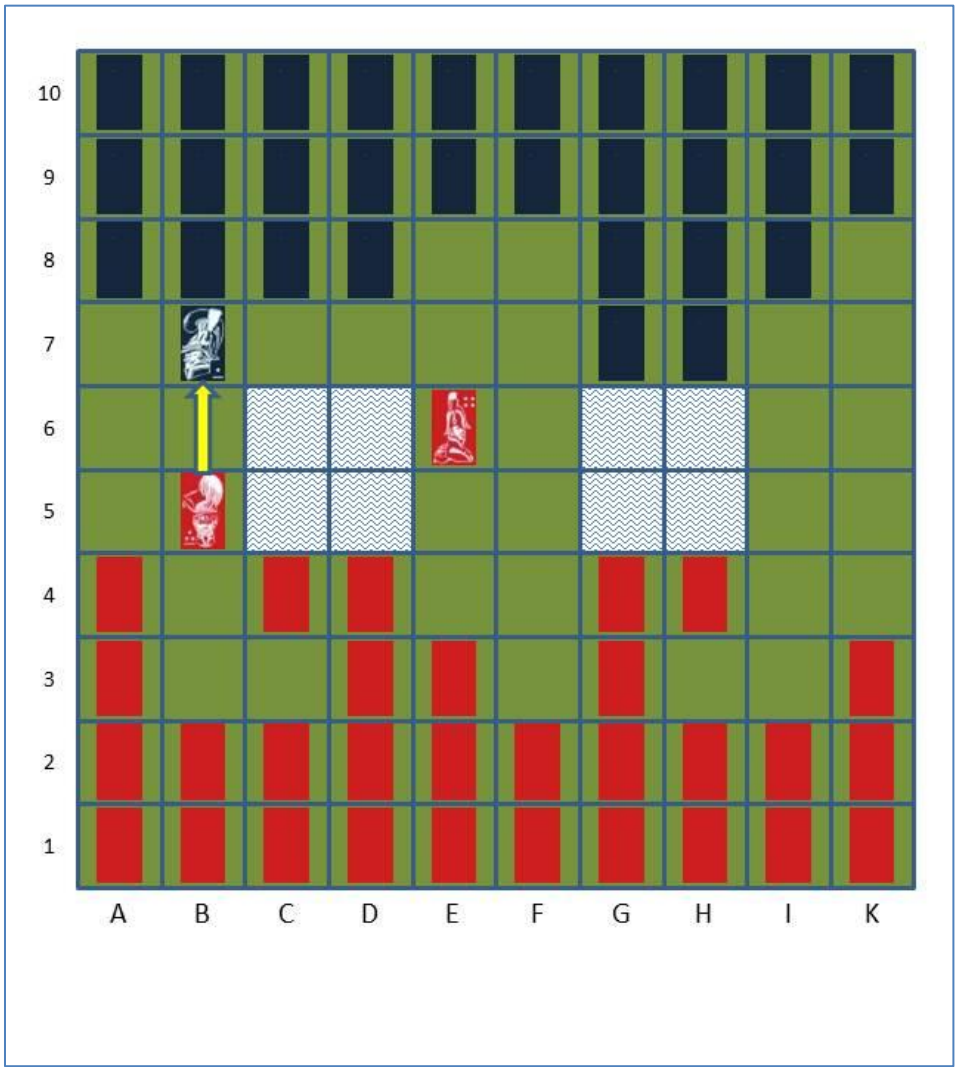


The sideway mobility of the blue miner is restricted to two columns.
The two-squares rule is in force. The miner will be captured by the sergeant.
This kind of capture can be foreseen and is suitable for the implementation in a static look ahead algorithm.



The sideway mobility of the miner is restricted to two columns.
The two-squares rule is in force. The miner can evade any attack of the sergeant.
This kind of evasion can be foreseen and is suitable for the implementation in a static look ahead algorithm.

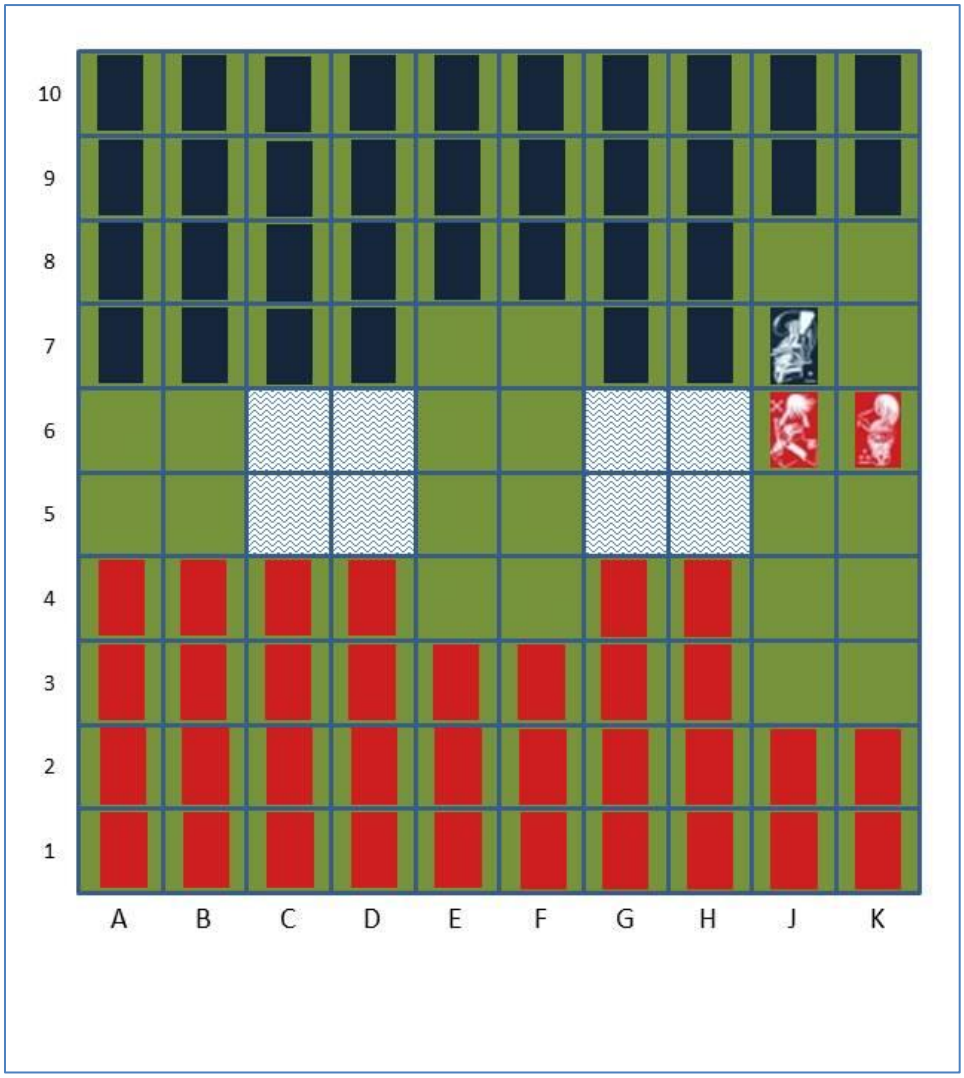
B.2.4 Encirclement



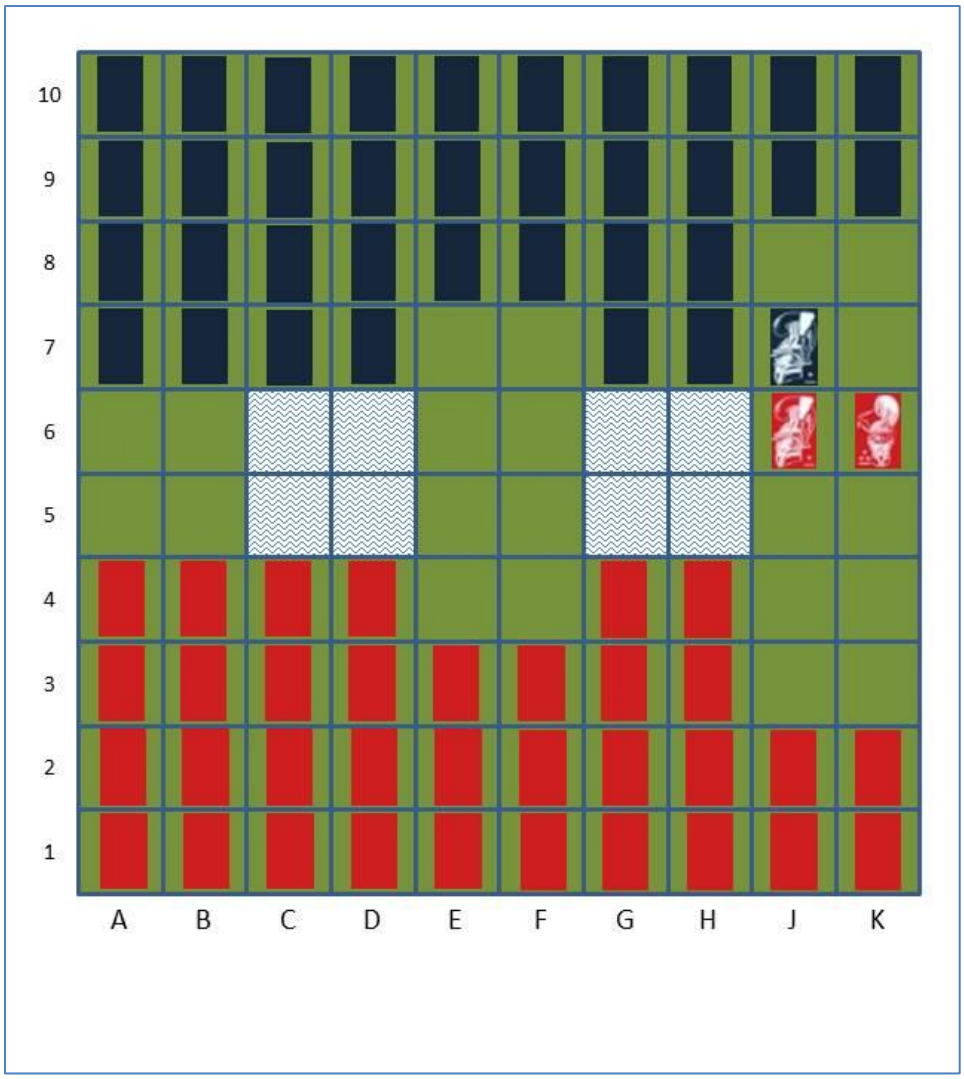
The red colonel can chase the blue major to E7, where the red general can capture the major.
This kind of chase works fine if the red general and colonel are invincible. If not, then this encirclement probably is too risky for the red colonel.
This kind of capture and its risk can be foreseen and is suitable for the implementation in a static look ahead algorithm.

B.2.4 The bulldozer attack

This is a special variation of encirclement, but it is interesting because of its bluffing possibilities.



The blue major has a known rank, but the red marshal and colonel do not have a known rank.
The blue major will be captured.

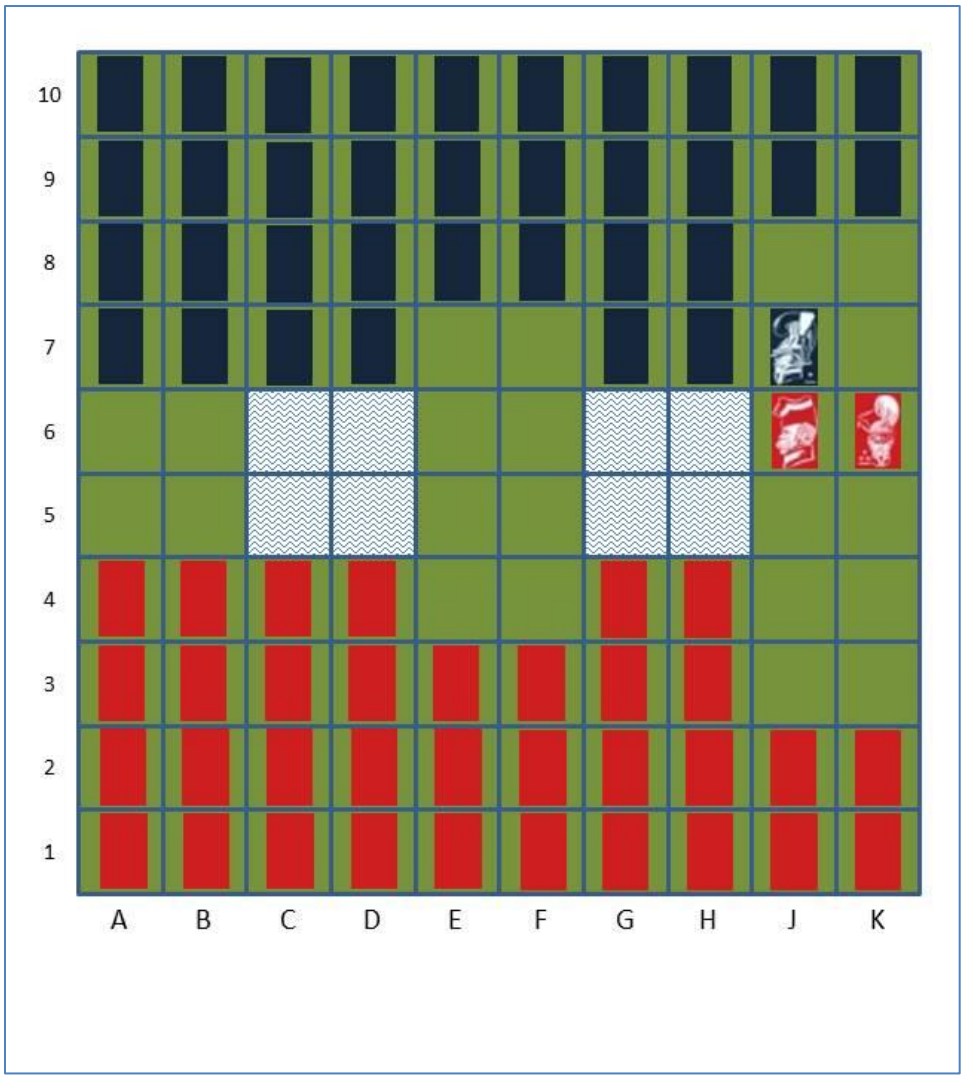


In this position the blue major has a known rank too.
The red attackers on J6 and K6 do not have a known rank.
For blue this position looks the same as the previous position.

This shows a dilemma for blue. Red may bluff, but which red piece should be considered as a real attacker and which not?

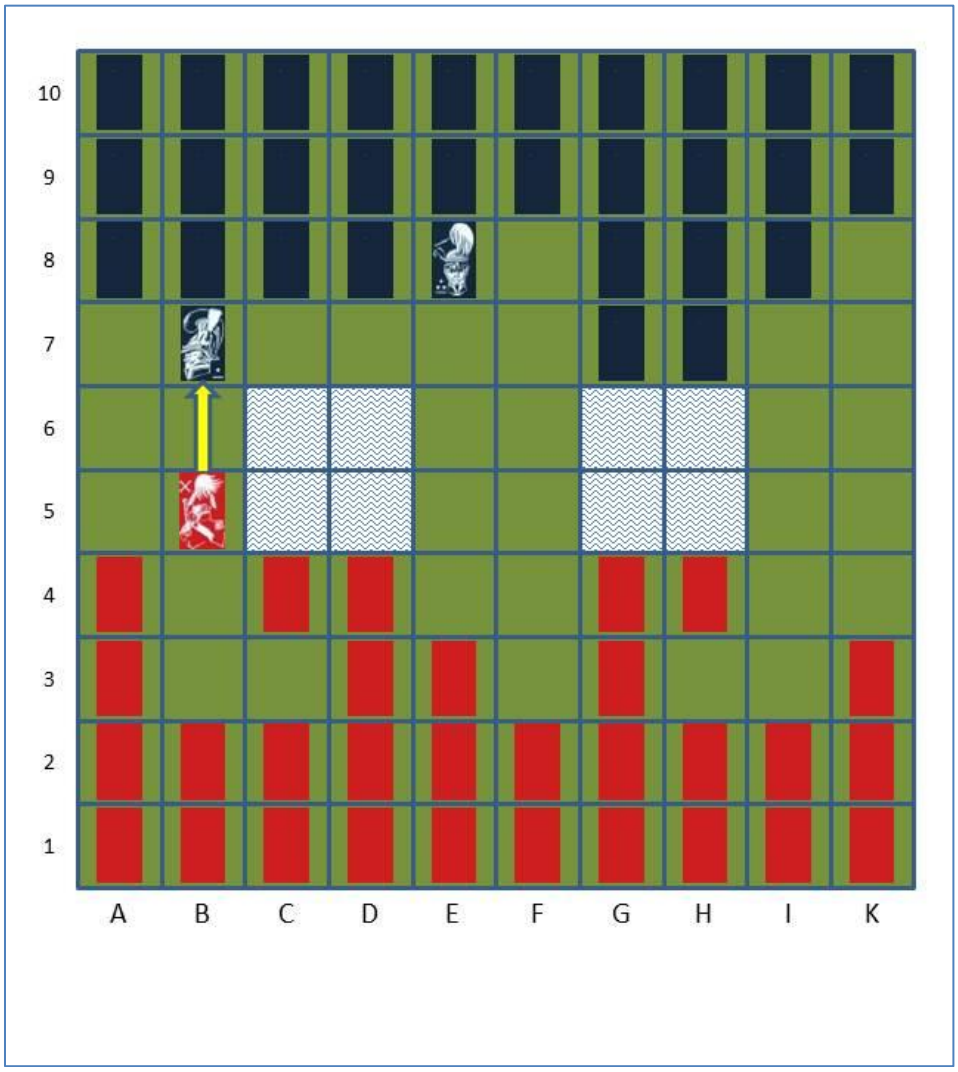
This kind of bluff can be used if a piece of the opponent cannot be conquered by the two-squares rule. It has a 50 % probability of success.

It may seem risky to expose the red major to high ranked blue pieces on H7, H8 or J9 but blue should consider the probability that the red piece on J6 has the rank of marshal. This kind of situations offers a though challenge for decision making to both human and computer players.



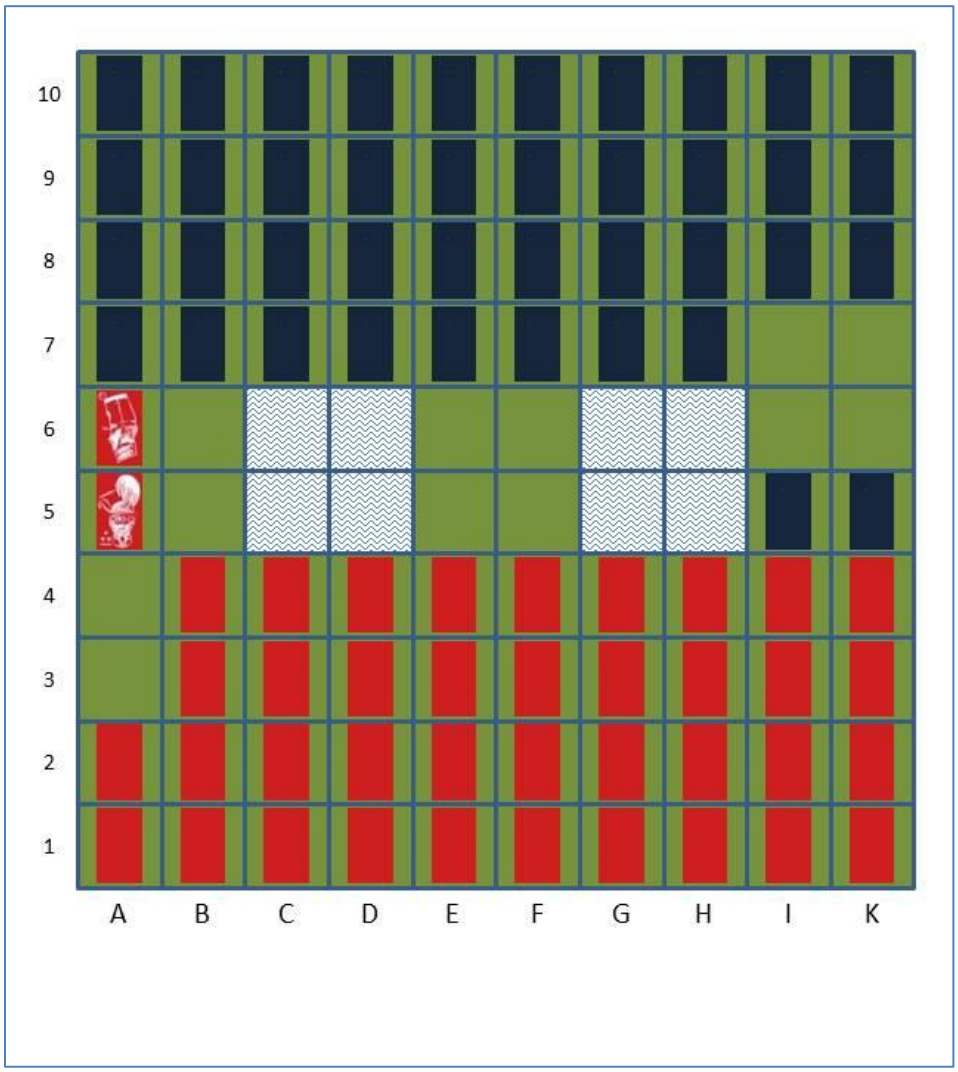
This is the real thing.
Red hopes that a bluff with a sergeant will lead to the capture of the blue major. The reward of winning a piece by a successful bluff is enough to accept the 50 % probable loss of a sergeant.

B.2.5 Chase with a fork



The red marshal can chase the major over E7. At E7 the marshal threatens the capture of both the blue major and the blue colonel.
This kind of chase works fine if the red marshal is invincible. If not, then this chase probably is too risky for the red marshal.
This kind of capture and its risk can be foreseen and is suitable for the implementation in a static look ahead algorithm.

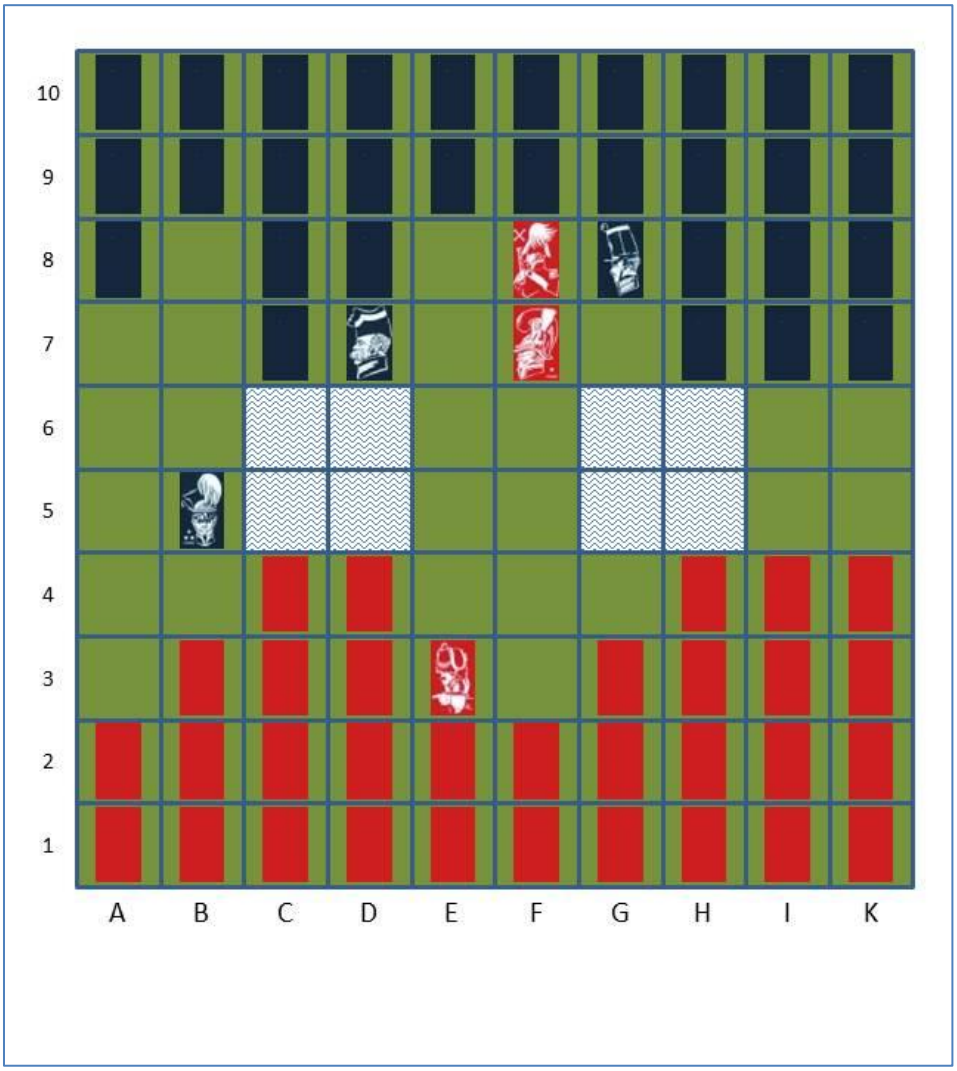
B.3 Explore and conquer



A cooperative unit has been sent into the left blue area. The captain will explore B7 and if a blue major captures the captain then the blue major will be captured by the red colonel.
There are some risks, but teams like this one gain material so often that on a par the result is positive. Some other combinations of a low and (middle) high rank too have more success than failure.

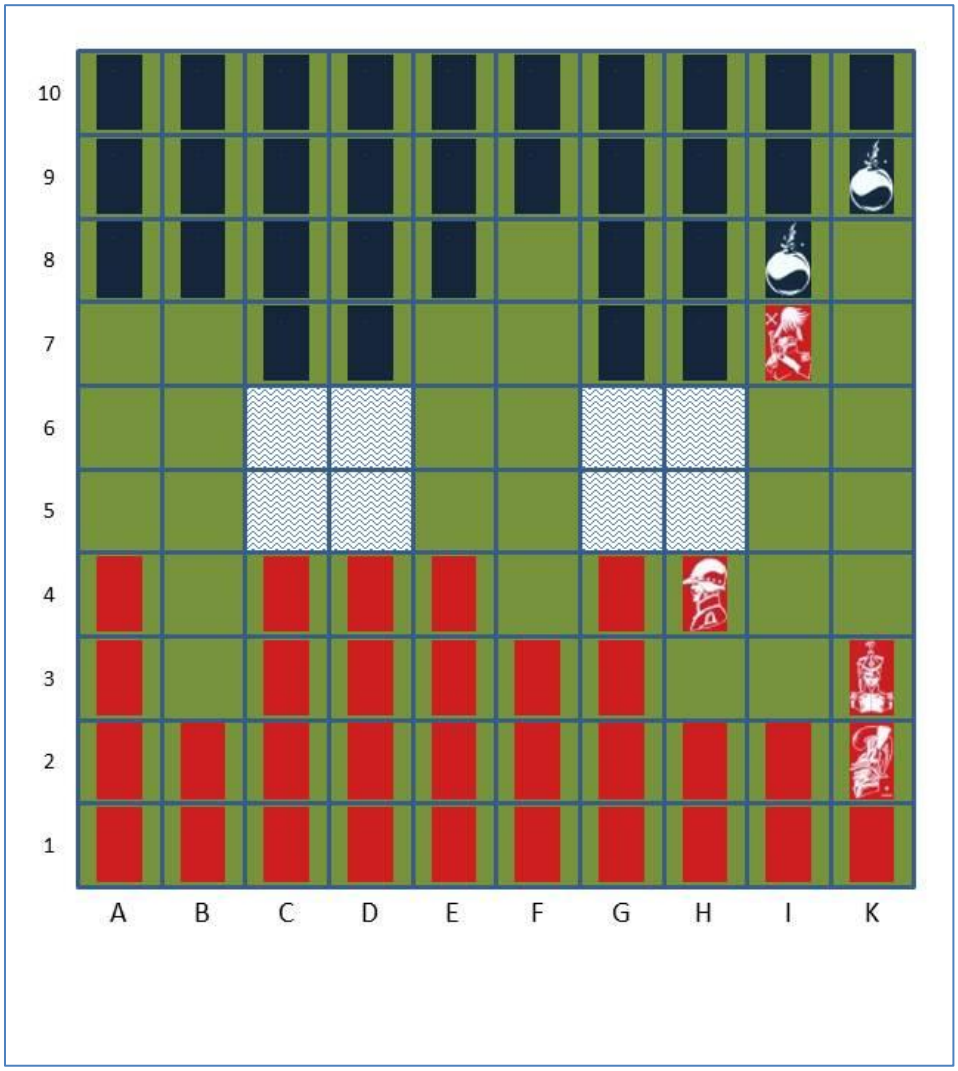
Here the higher rank keeps close to the exploring piece, but it also is possible to keep the colonel on its initial position and move only the captain into the blue area. Many patterns are possible. Human players use much variation in move pattern and team contents. Often a scout is used and mimics the behaviour of the lower or higher rank in a team. The intention of this bluff technique is to provoke and thereby to discover a high rank of the opponent.

B.4 Control in an area

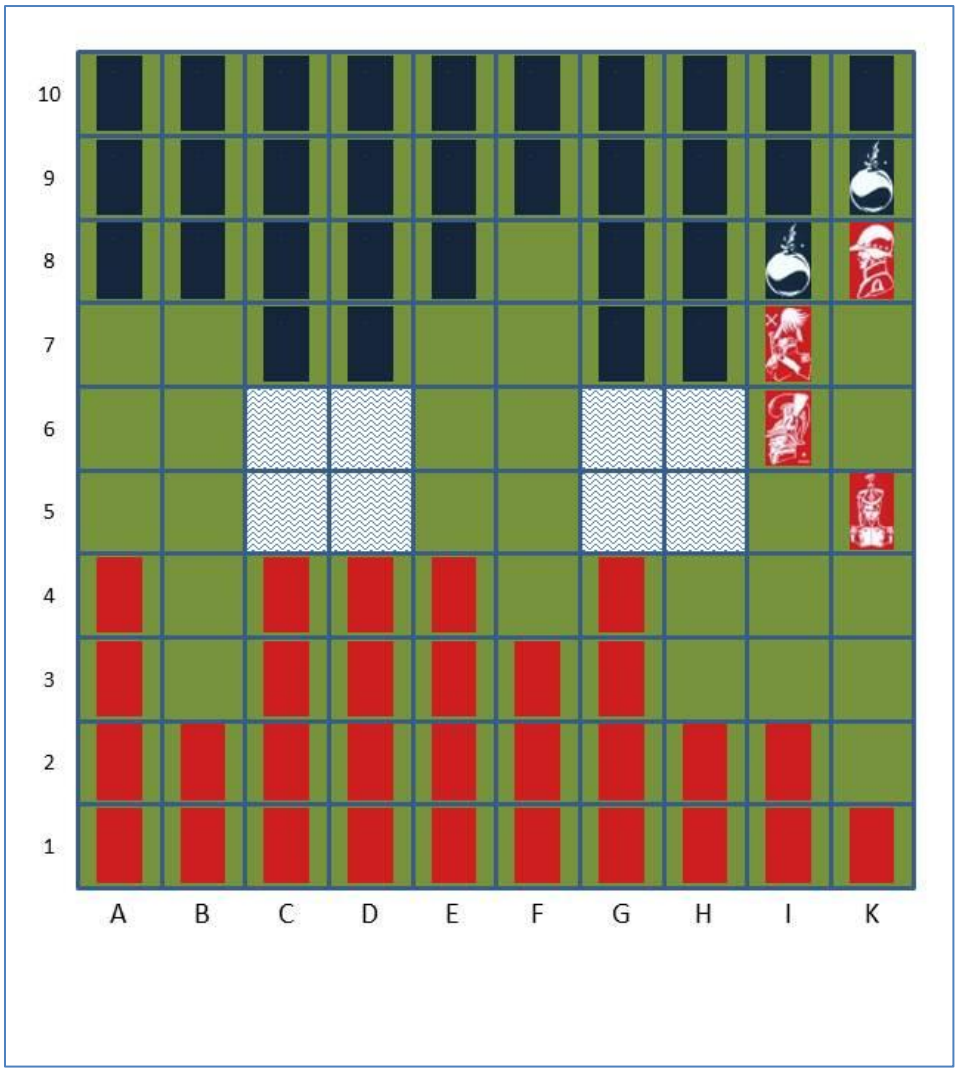


The red marshal has been placed with some risk on F8 and enables on that position the march of lower ranked pieces over the E column. Note that some blue pieces with lower ranks (the sergeant on D7 and the captain on G8) are not captured, because they block the defence. The long-term goal is to conquer the E column and the tenth row.
If necessary and meaningful a good playing opponent may sacrifice blocking pieces in order to open connections for the defence against this strategy.

B.5 A siege

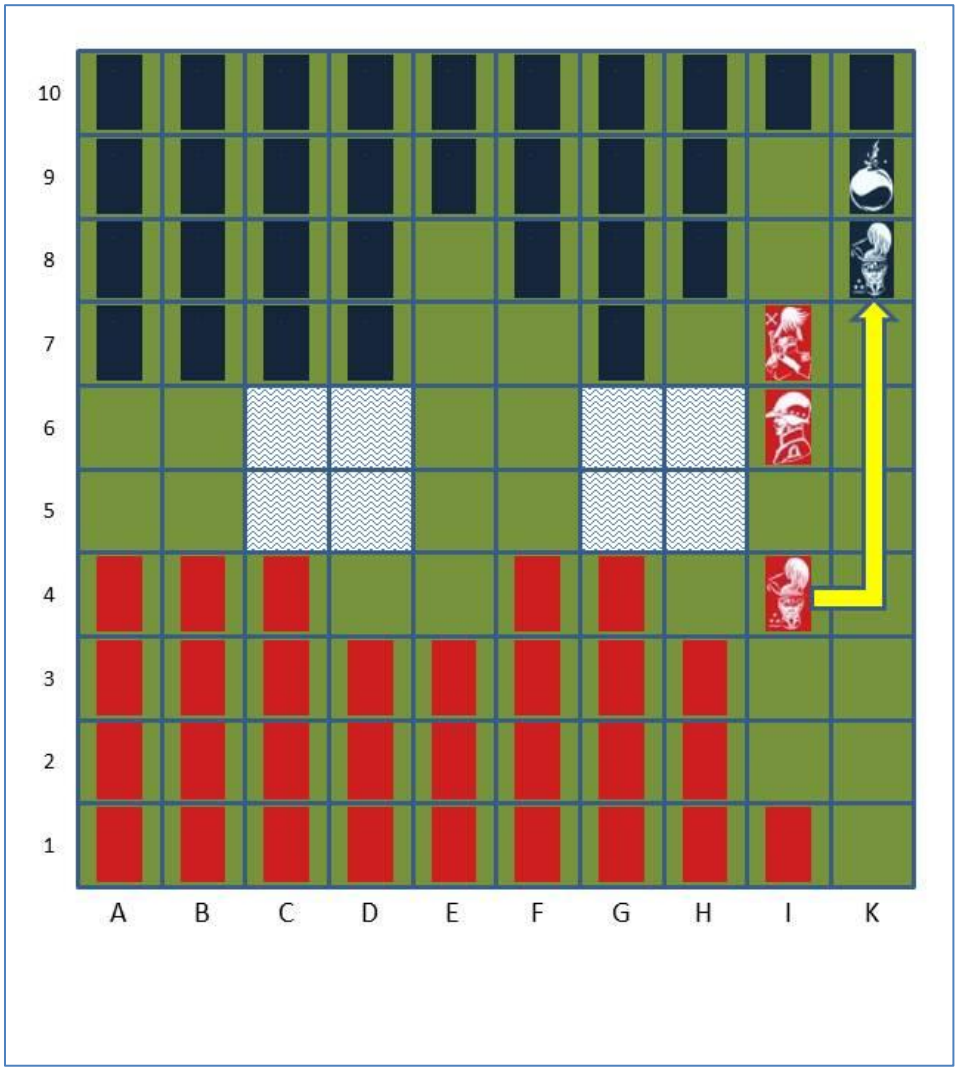


After the discovery of a bomb barrier an attack can be arranged to the area behind this barrier. Here the main target is the bomb on K9. The miner on H7 has to be moved to K8. When the miner has arrived there is no need for hurry. It's a good practice to postpone the capture of the bomb and to move other pieces as near as possible to the target area.



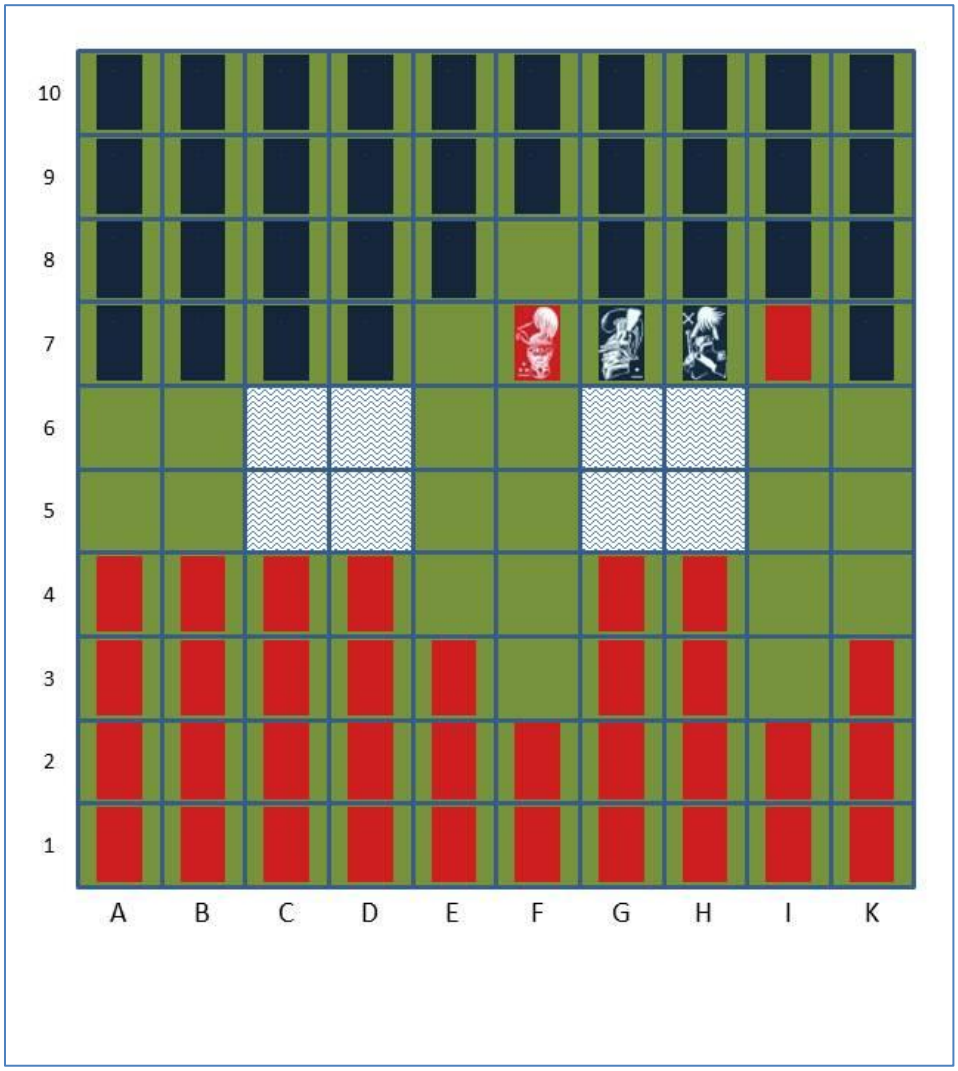
This is the ideal preparation of a capture on K9. If some blue piece captures the miner on K9 then a choice out of three ranks has been enabled by the groundwork done before the breakthrough.

B.7 Exchange of a defender



After the discovery of a bomb on K9 the attack on this bomb by a miner is being blocked by a blue colonel on K8. The two-squares rule prevents the capture of this colonel by the red marshal. The blockade can be removed by an exchange of colonels. After the exchange the red miner can march to the bomb on K9.

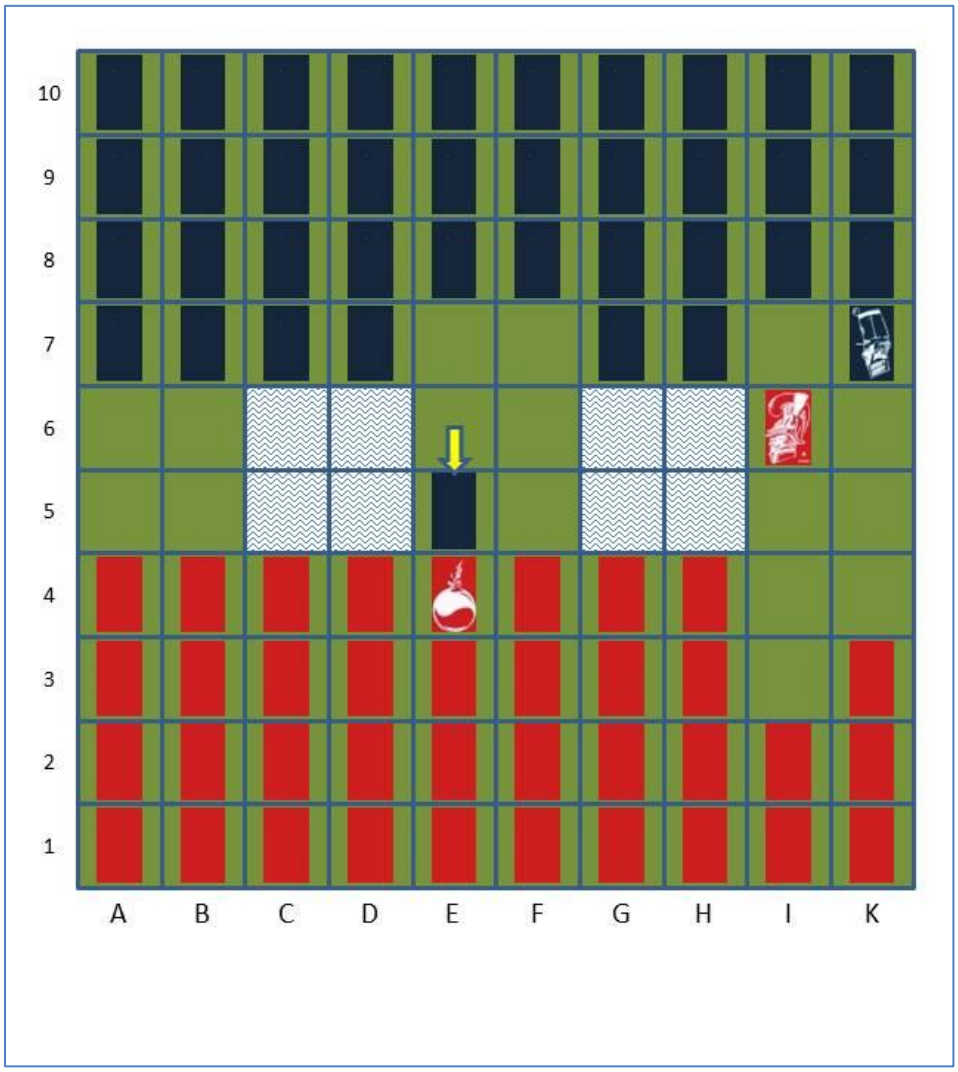
B.8 Bind a high rank



If the blue marshal leaves H7 then the blue major on G7 will be captured. Therefore the blue marshal is tied to the blue major. Any red piece (maybe even the spy) may enter the square J7 with little risk to be captured by the marshal.

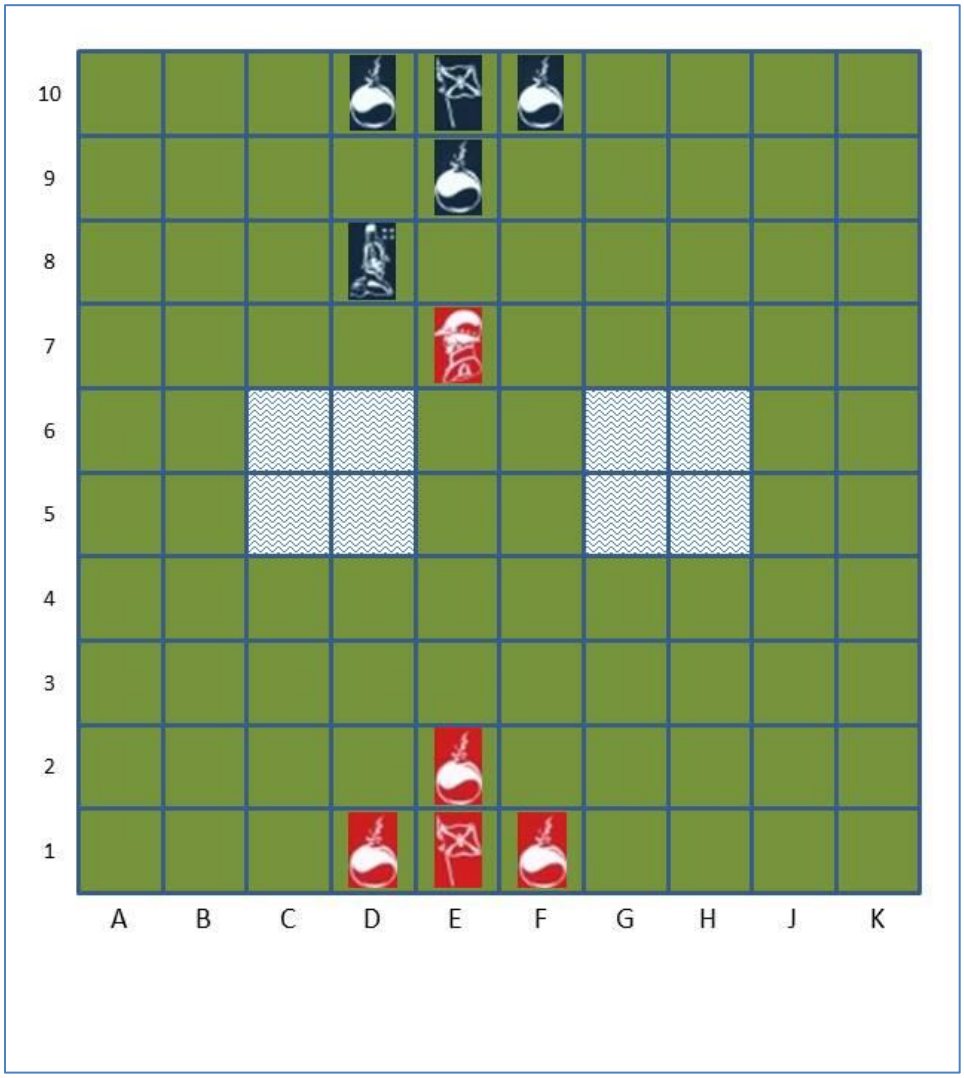
B.9 Stalking at the right moment

This is a “technique” only to be used against human opponents.



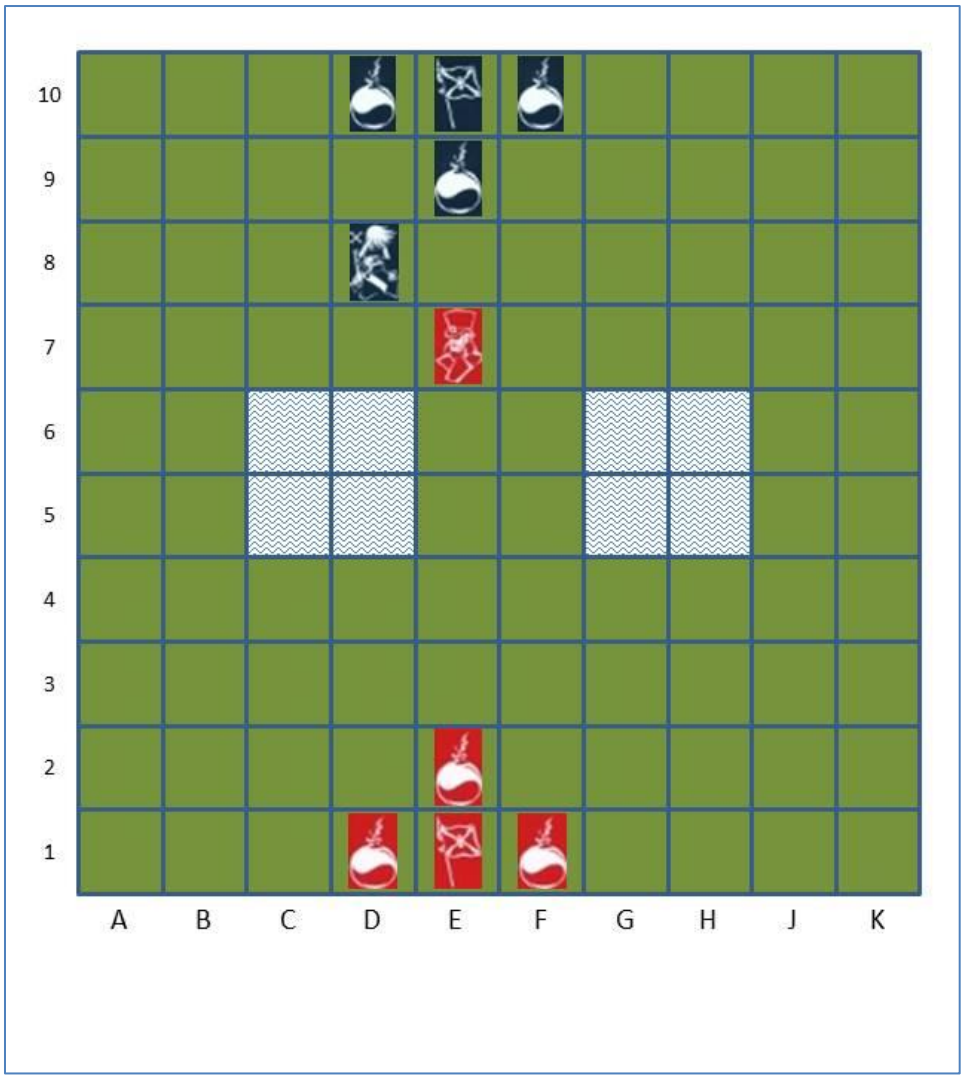
The known red major cannot capture the known blue captain, because the blue captain can evade threats by use of the two squares rule. But maybe the captain may be won if the blue (human!) opponent is busy with an action. Often a human player concentrates too much on opportunities in the own actions.
In this example a red bomb has been discovered and the last blue move was E6-E5 with an unknown piece that (very) probably is a miner. The next move of blue almost certainly will be E5-E4.
That’s the right moment to do a speculative move with the red major to K6. If the blue captain evades this threat then the chase of the captain should stop. But very often a human player’s attention will be focused too much on the current move to E4 and then the blue captain will be captured.

B.10 Forced moves



In the end game forced moves are an important instrument to win a game.

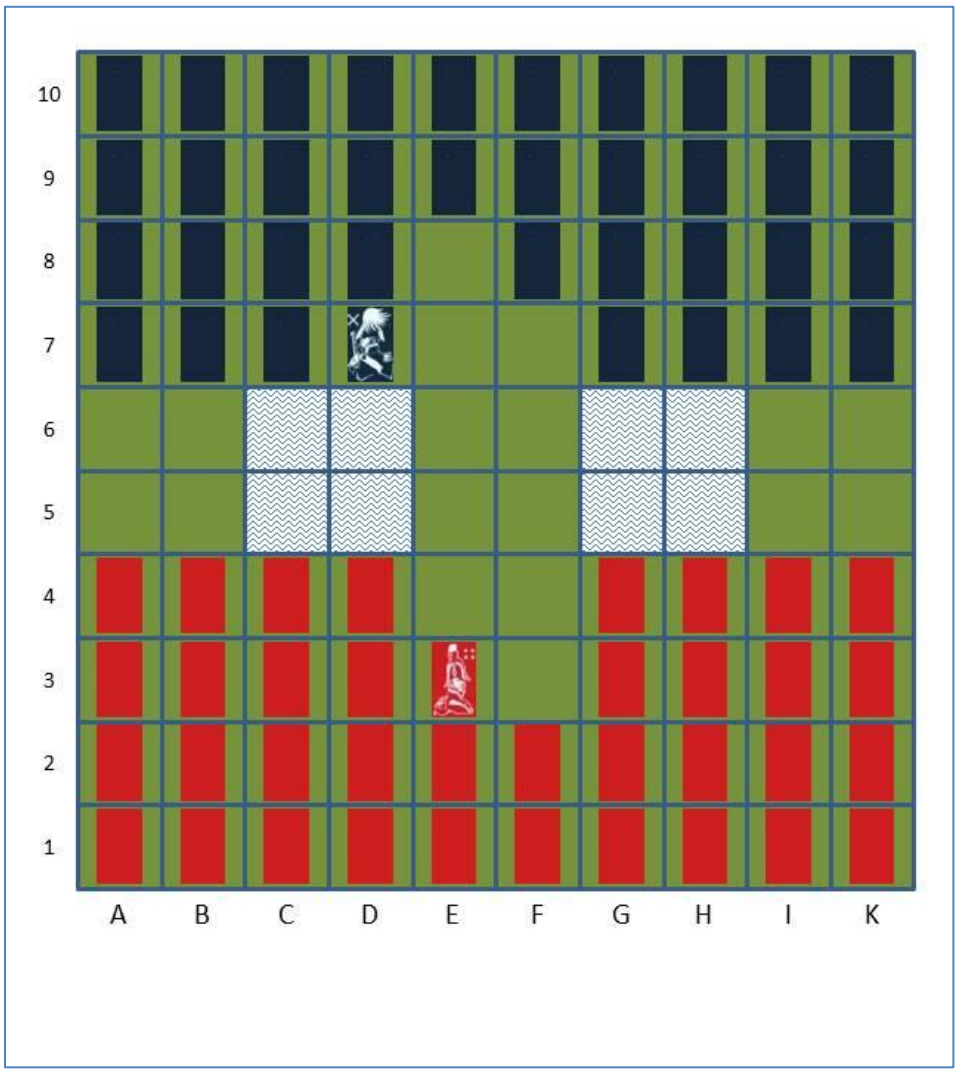
In this example blue has three bombs, one flag and a general.
If red is in turn then the game is a draw. But if blue is in turn then red will win because blue can only make forced moves. The two-squares rule forbids a continuous defence of squares that offer access to the blue flag. Finally the general has to concede a free path to the blue flag.



The same principle is in force here. If red is in turn then the game is a draw. But if blue is in turn then the red spy can force the blue marshal to a smaller area until there is no escape anymore.

Appendix C: Some examples of defensive goals

C.1 Evasion of a threat

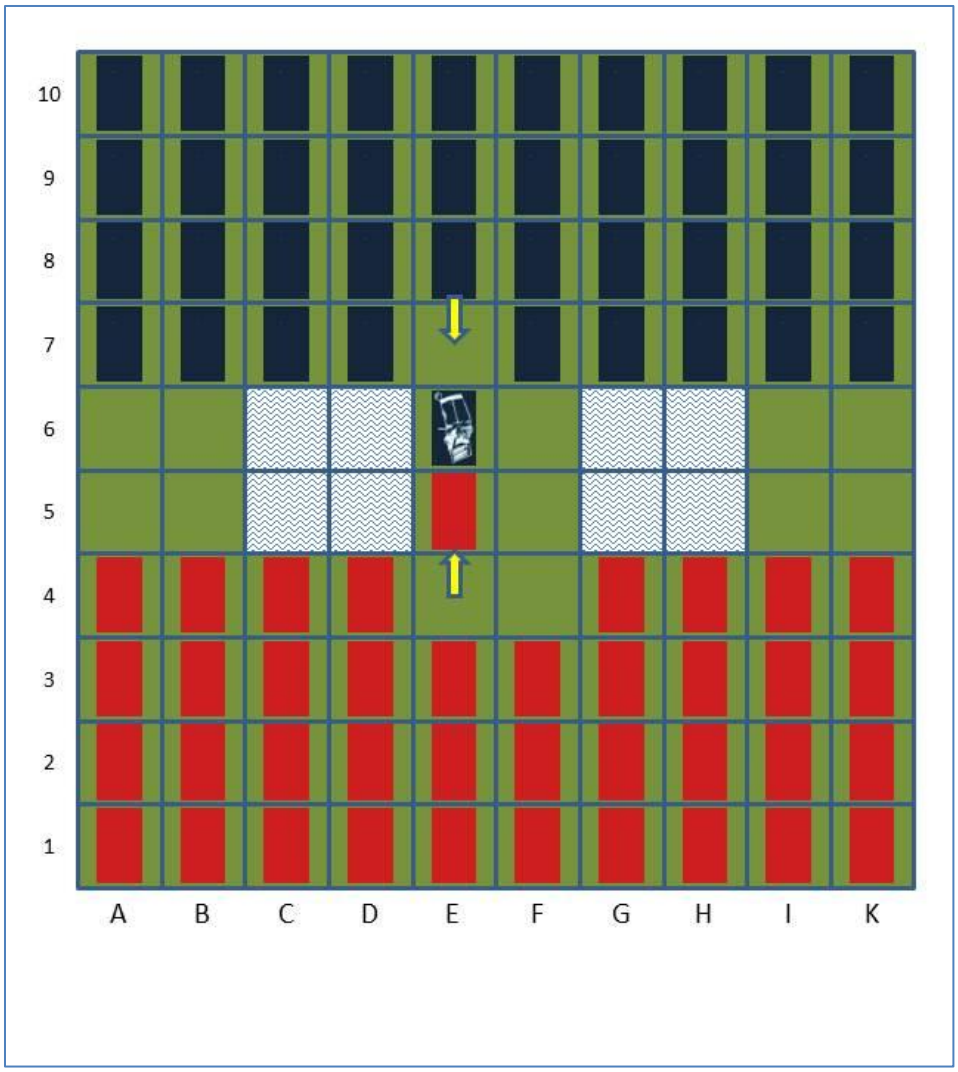


If the blue marshal moves to E7 then the red general cannot evade the capture by the blue marshal because the two squares rule forbids a continuous evasion. Therefore it is necessary to move the red general to F3. Then the two squares rule will prevent the capture of the red general by the blue marshal.

This kind of capture and its risk can be foreseen and is suitable for the implementation in a static look ahead algorithm.

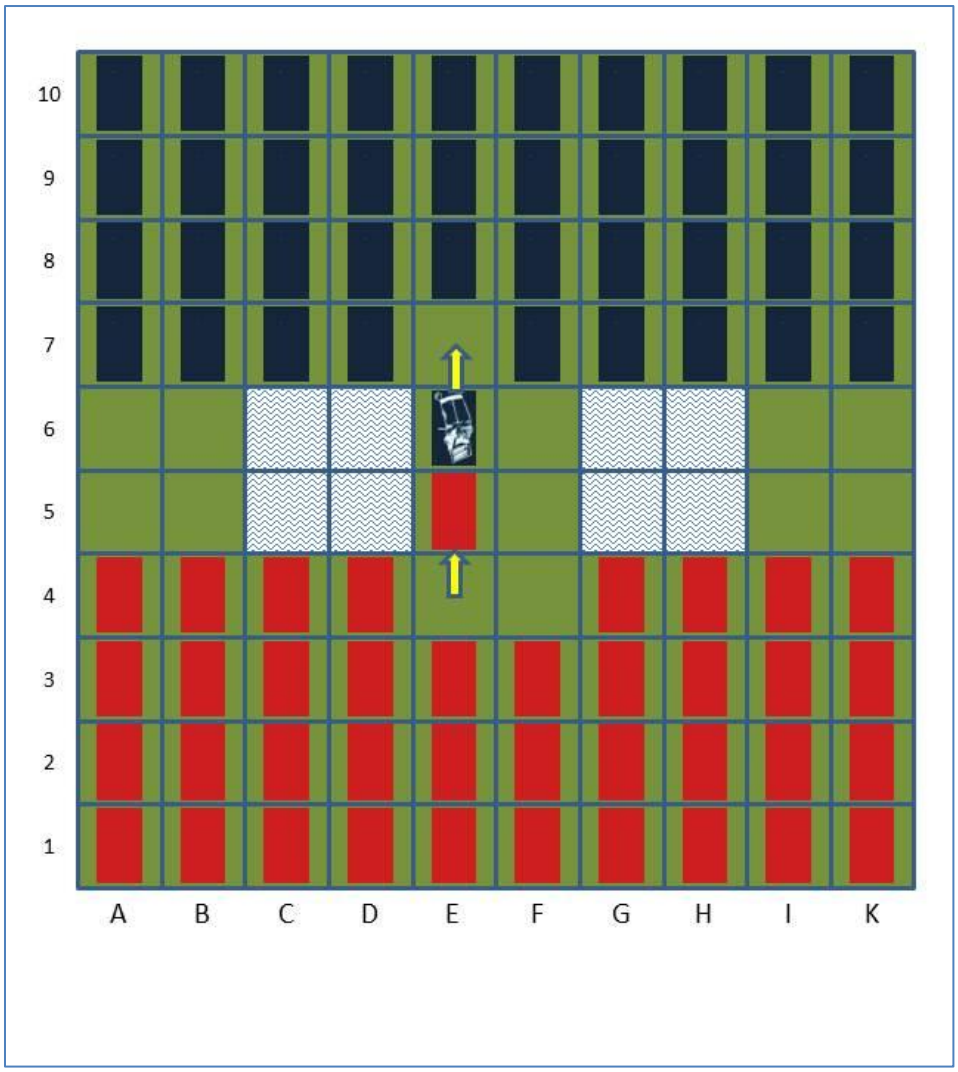
C.2 Protection of a threatened piece

Some trivial examples of protection follow here.



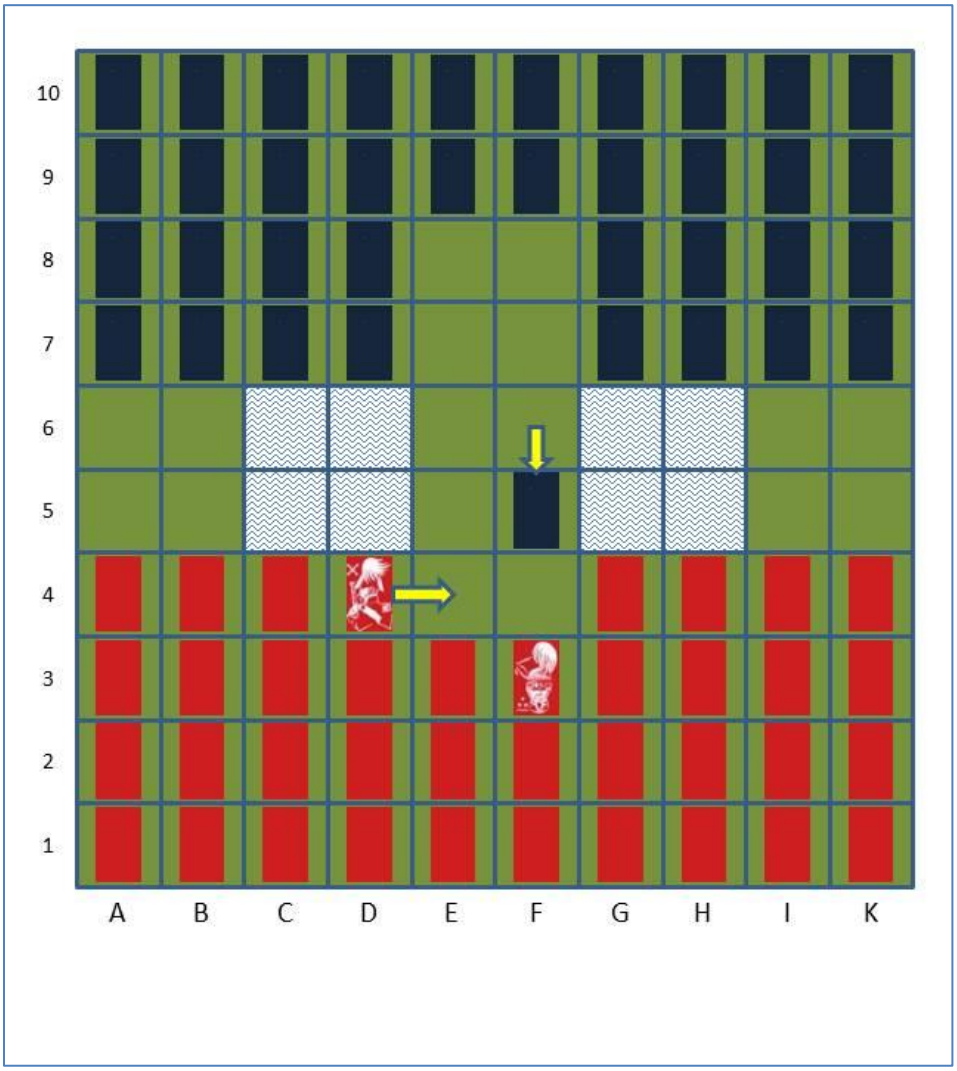
The blue captain may be under attack by the red piece on E5 or maybe this only is an illusion of attack. If an unknown blue piece moves to E7 then it is supposed to protect the blue captain by the high rank it may have.

Nothing is certain. This makes Stratego especially attractive.



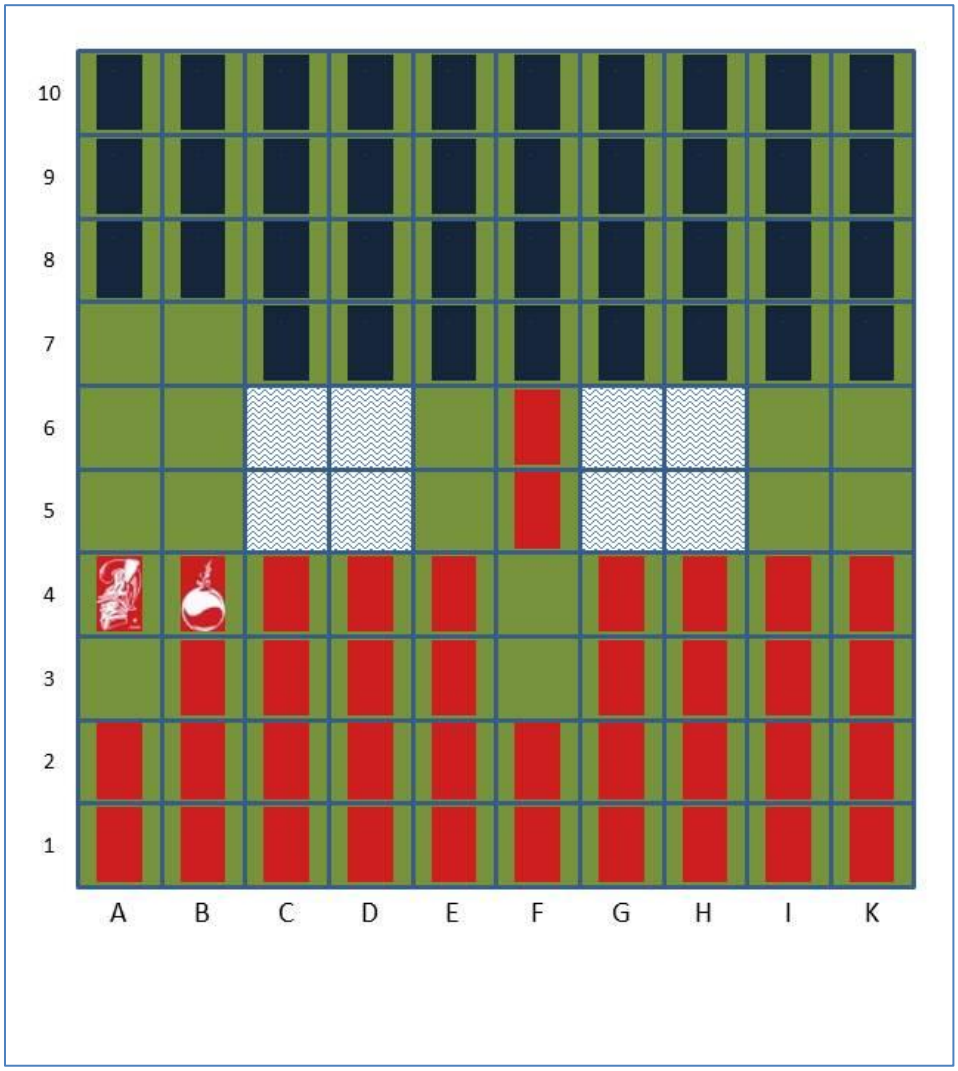
Again an unknown red piece has been moved to E5 and tries to convince the opponent that the blue captain is under attack. If the blue captain retreats to E7 it may find a resort there, but it is possible that on that square it too has the same amount of protection as on square E6.

C.3 Interception of the attacker

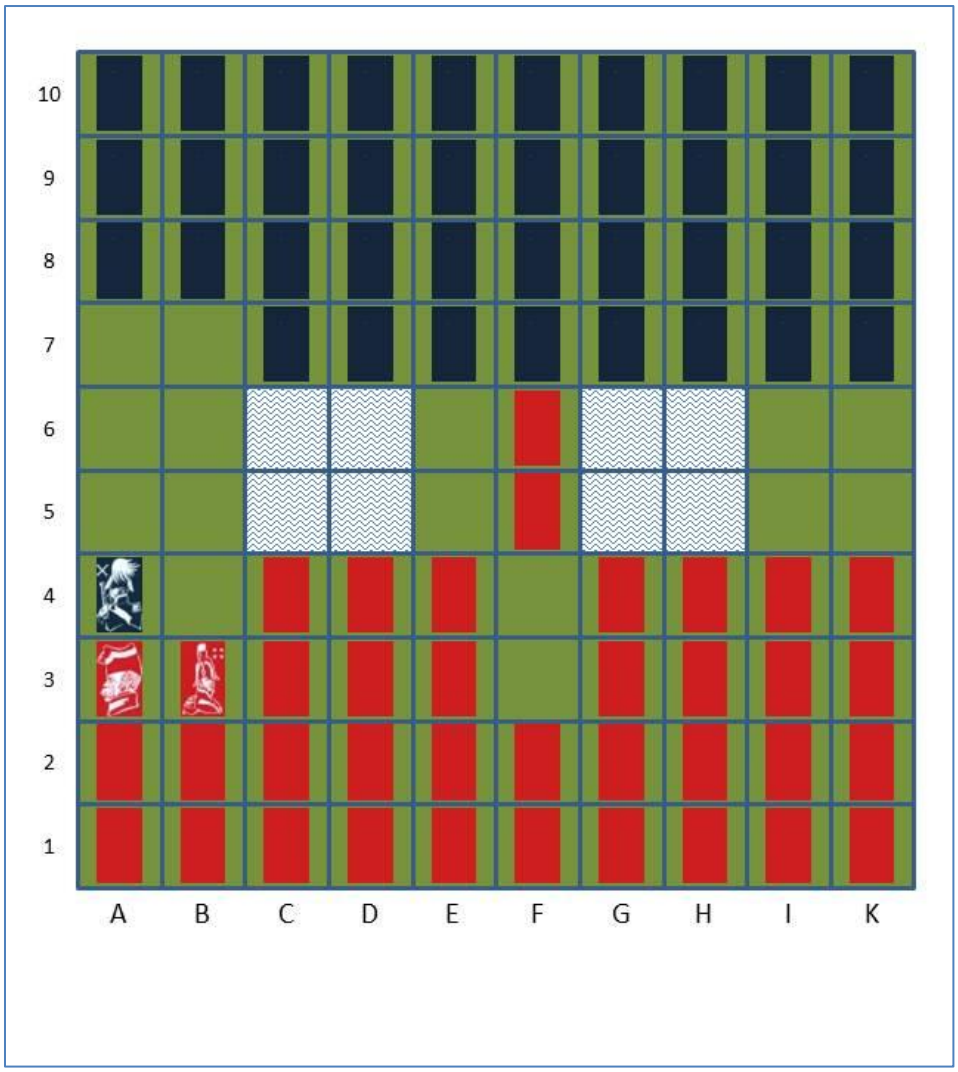


The unknown blue piece on F6 moves to F5. This looks as a serious threat to capture the red colonel on F3 but too it may be a good attempt to discover the position of the red marshal by a blue scout. Hopefully the move of the red marshal to E4 really is necessary; else the exposure of the rank has been forced at a minimal price. If the blue opponent attacks with a general and does believe the move to E4 to be a bluff then red is having a party in this game.

C.4 Mobilize potential targets



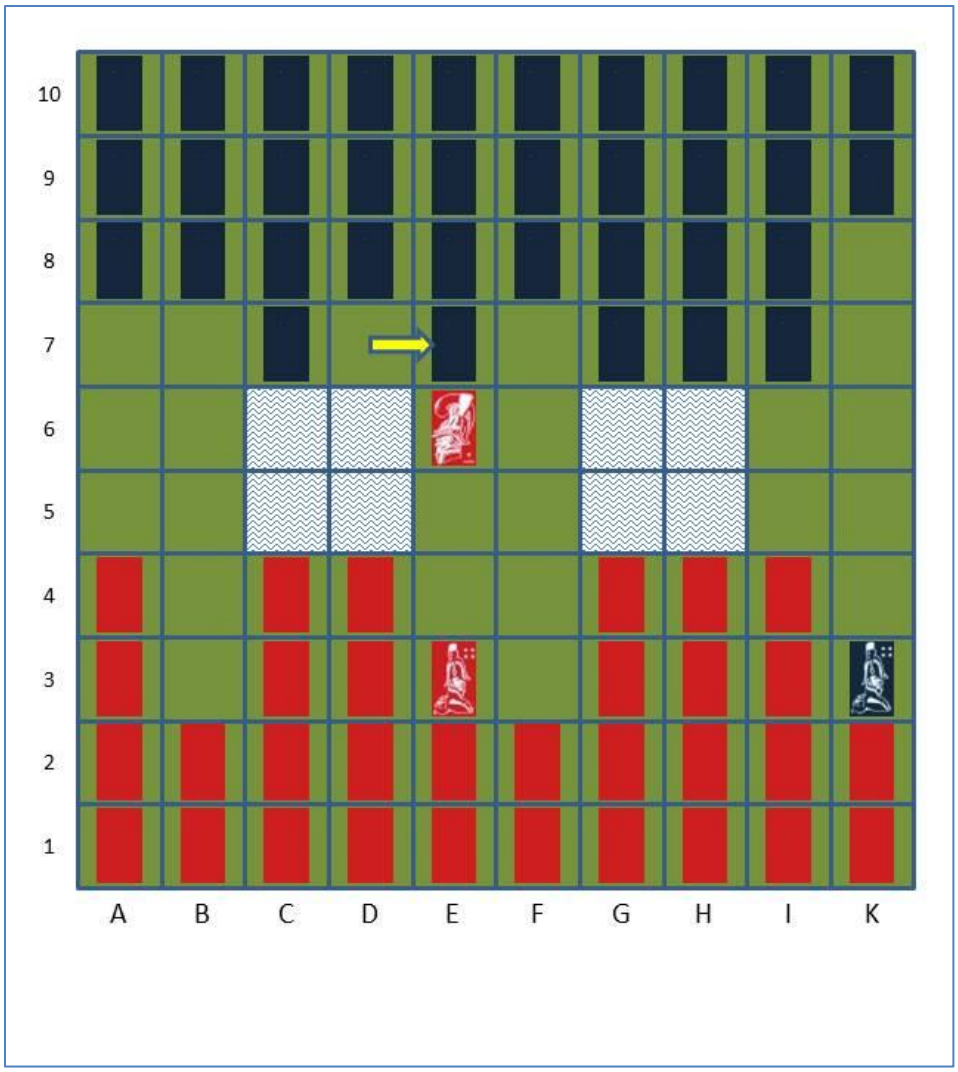
The rank of the major on A4 has been discovered. If blue has a colonel or higher on A8 then this major will be lost. But if a blue colonel or higher has to come from the B column then the red major may be saved if sideway moves between the A and B column are enabled. Therefore the red major has to be moved to A5.



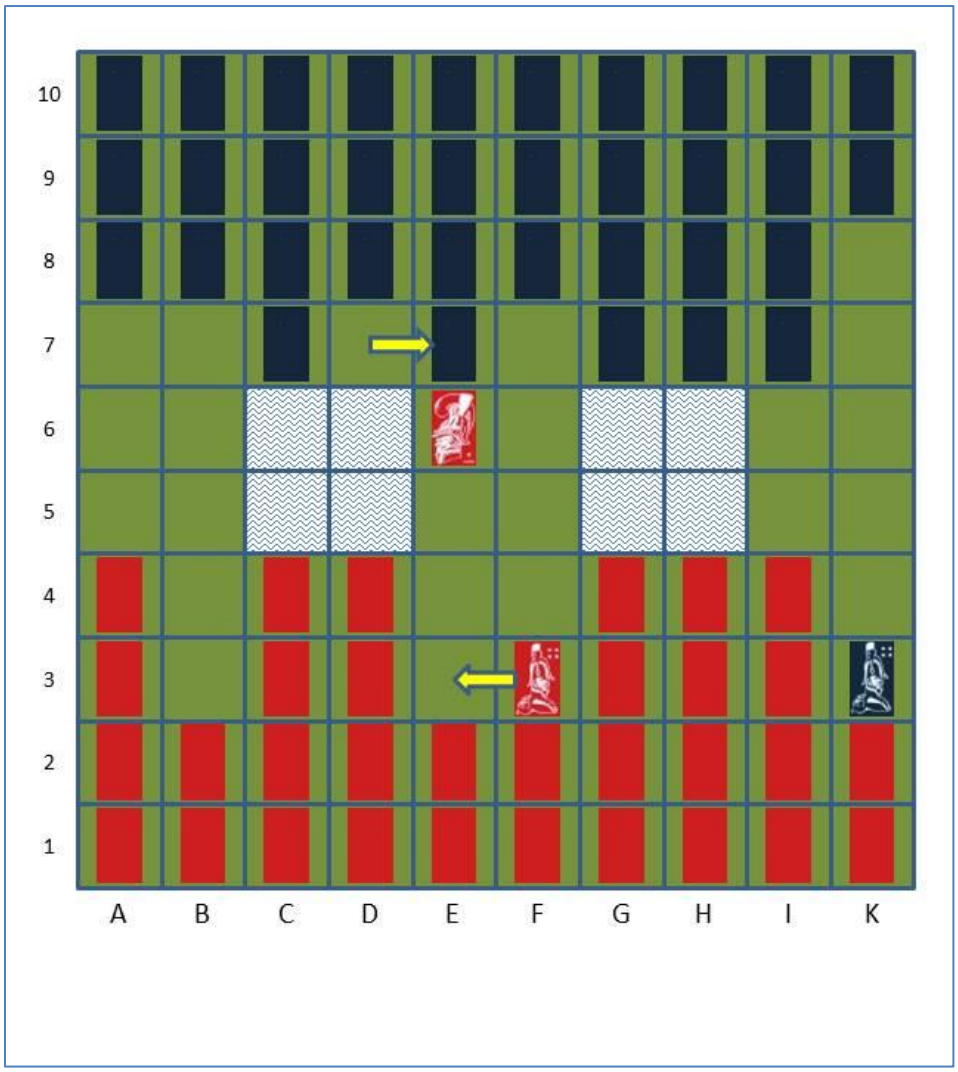
The red general has been exposed and the blue marshal threats to capture this immobile piece.
If red sacrifices the red sergeant then the red general gets sideway mobility on the A and B columns.
Then the two squares rules will prohibit the capture of the red general by the blue marshal.

C.5 Long distance protection by the two-squares rule

Protection can be given by a high ranked piece at a distance.

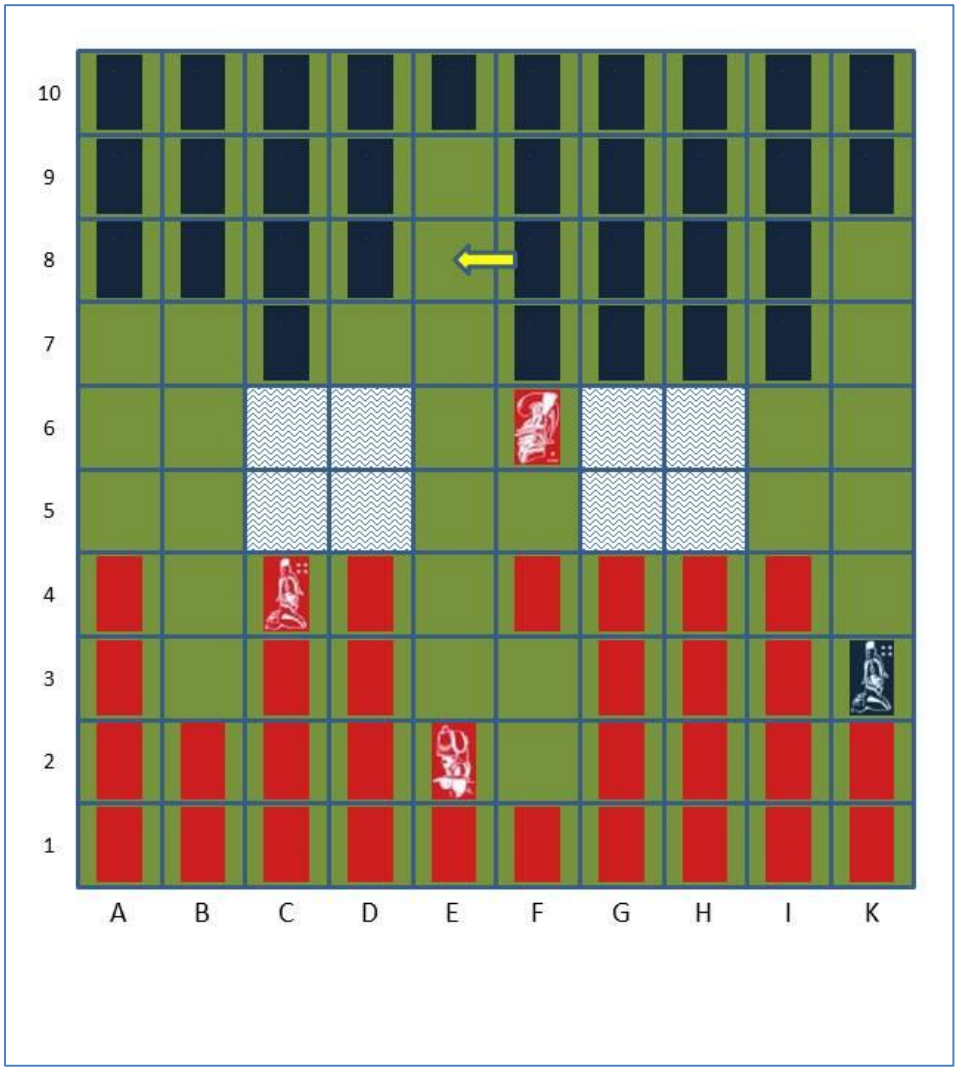


Assume in this position that the marshals have been exchanged. The generals are known and invincible. An unknown piece on D7 moves to E7 suggesting an attack on the red major. This attack will fail because the red general will be able to conquer a blue piece on E6 by use of the two squares rule.



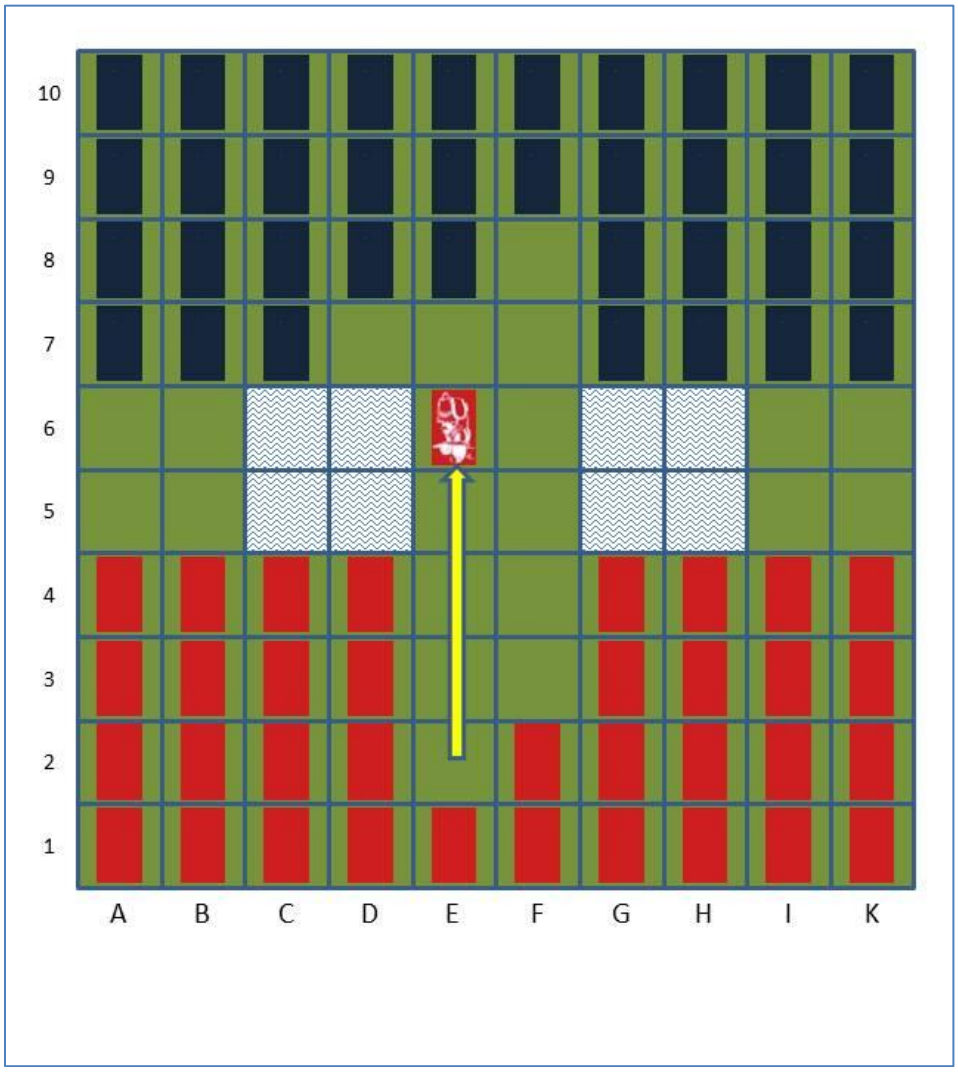
This position is almost identical to the previous one.
Here again a piece has been moved to E7 and threats to capture the known red major.
A move of the red general to E3 will enable the red general to conquer a blue piece on E6. So this will prevent the capture of the red major.

C.5 Explore the rank of a potential attacker



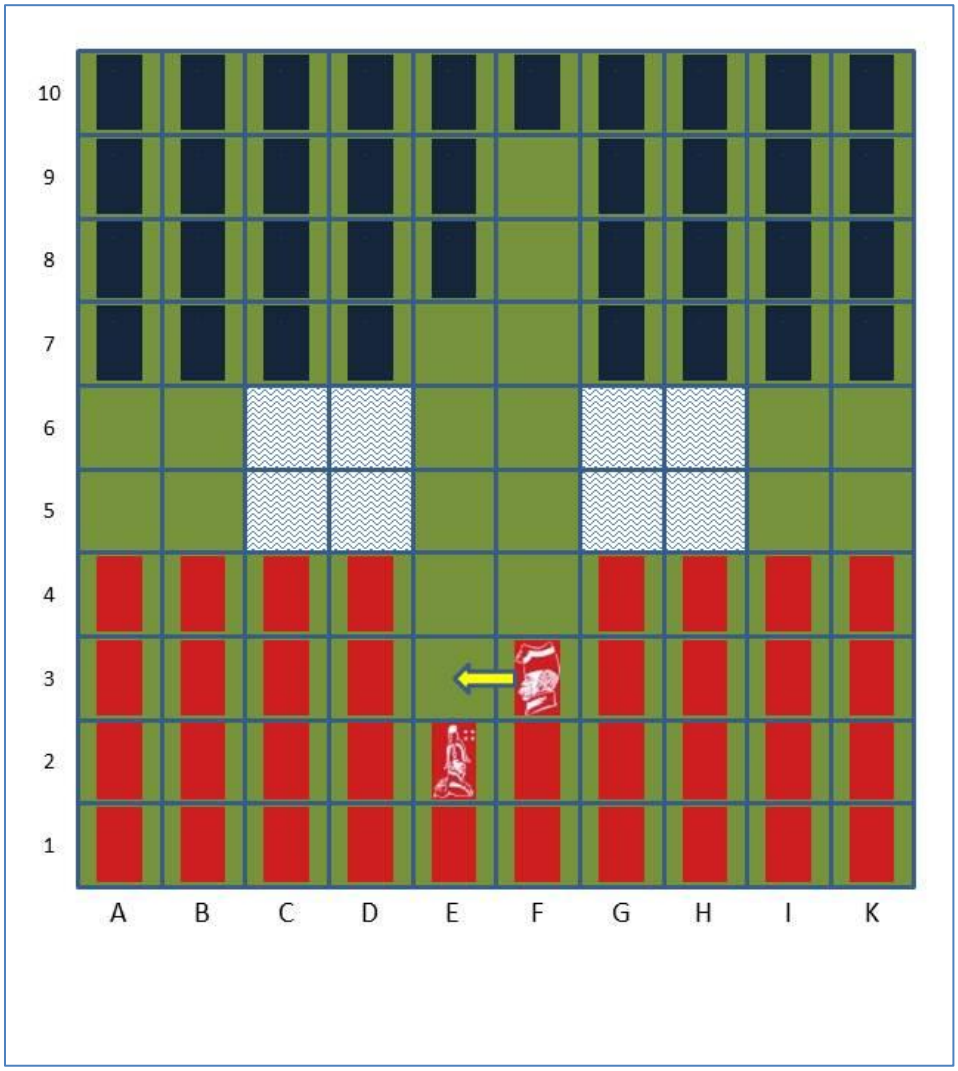
In this position the generals are known and invincible.
The last move of blue is F8-E8 with an unknown piece.
It is interesting to know what the rank of the unknown piece on E8 is. Use the scout on E2 to discover the rank of the piece on E8. Then red knows what to do with the red major. If the blue piece is a colonel then the major should retreat to F5. On this square the two squares rule will enable the red major to evade a duel with the colonel.

Detection may be enabled by placing a guardian piece on an access point



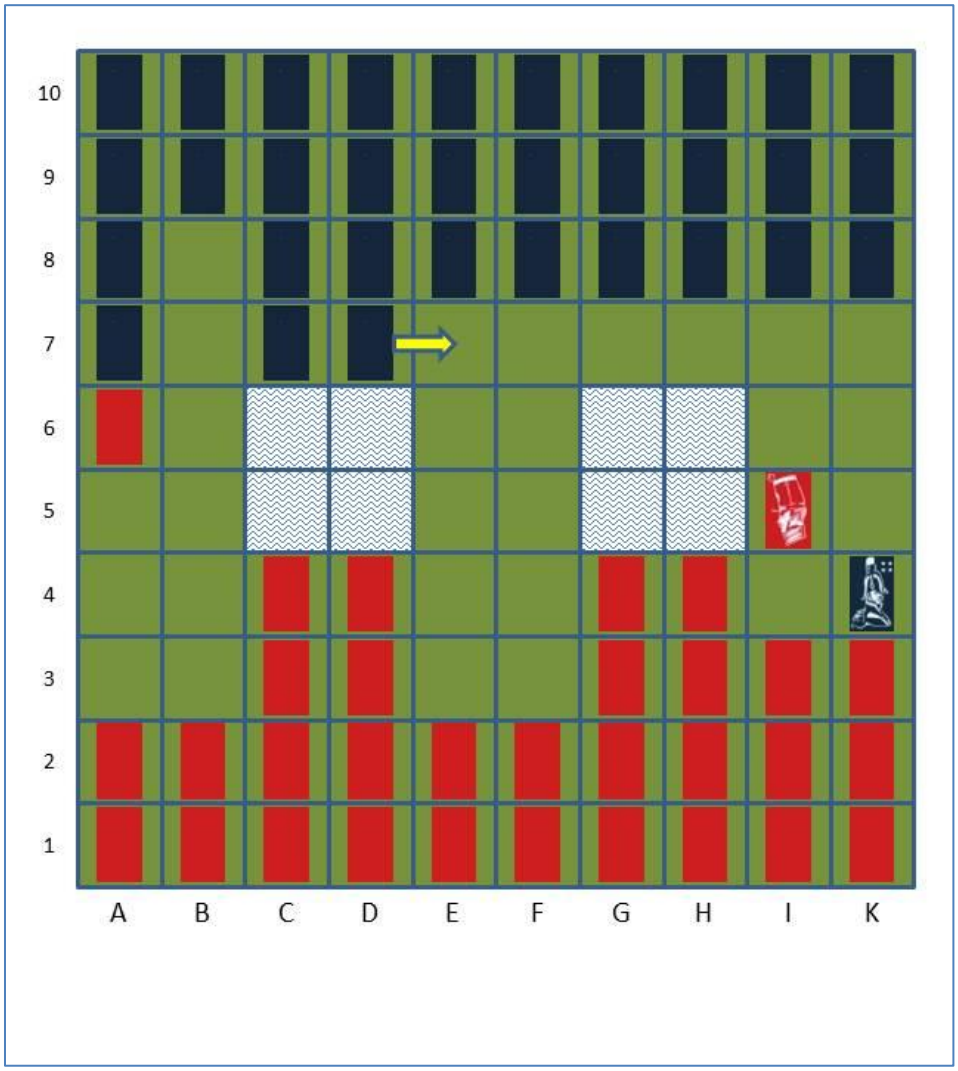
The scout functions as a guard to be sacrificed when any blue piece tries to enter the red middle area.

C.7 Prevent detection of a high ranked piece



A move of the sergeant to E3 will protect the red general on E2 for detection by a blue scout.

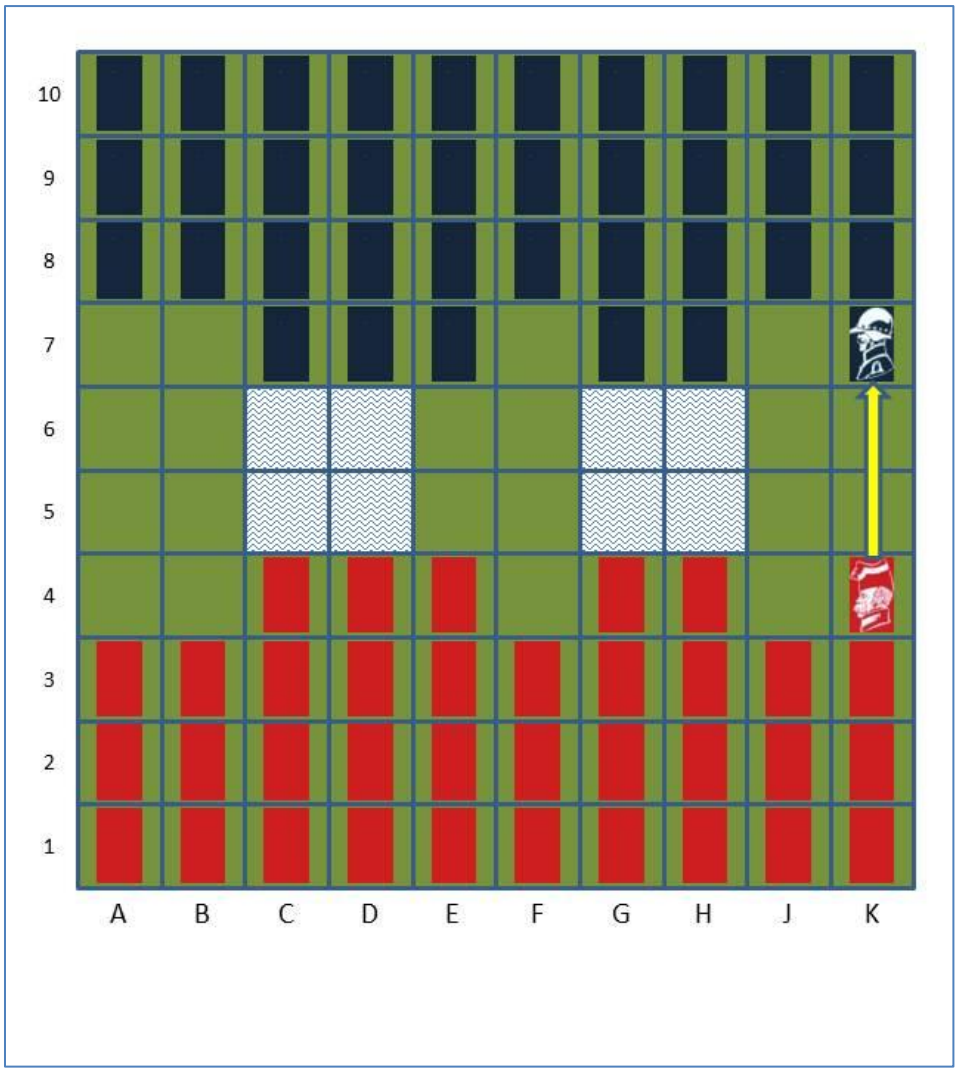
C.8 Defence by desperado attack



The blue general has expelled the red captain from I4 to I5. The next move of blue is D7-E7 by a piece with unknown rank. Probably (?, nothing is completely sure in Stratego) the goal of this move is the encirclement and thereby the capture of the red captain.
The red captain has no escape from such encirclement. In such cases the best policy is to get whatever still may be gotten and send the captain recklessly into the opponent area.
The captain is able to attain I8 just in time to enter the lines of the opponent and is willing to sacrifice itself for any scrap of information or material that may be gained by this manoeuvre.

Appendix D: Processing efficiency in local search trees

At first an example to illustrate efficiency aspects in local search trees.



In 24 moves the red sergeant on K4 is able to capture the blue miner on K7 by use of the two-squares rule. Brute force search trees of 24 levels require too much processing time. This example shows that a conventional brute force search will not work.

But fortunately in local search trees it is possible to reduce significantly the amount of computer processing. Here three techniques are discussed:

- Detection of goals by jump moves
- Selectivity by goal directedness
- Look ahead functions.

These techniques are heuristics that require game knowledge.

D.1 Detection of goals by jump moves

A substantial part of moves in a brute force search tree goes to an empty field that has no opponent piece next to it. The function of such moves is to decrease or increase the distance to a target duel. But in Stratego the real thing happens in duels. Therefore the only significant moves in a brute force search tree are the moves that place an own piece next to a piece of the opponent or engage into a duel. Moves to empty squares without a neighbouring piece of the opponent are irrelevant for the outcome of duels. Moreover they lead to irrelevant nodes in the search tree.

The removal of irrelevant nodes is a means to reduce the size of the search tree and to focus on the core of Stratego game logic. This leads to the idea of "jump moves". Allow pieces to jump to squares next to opponent pieces. Jump moves can detect goals over a long distance and can show what the outcome will be when pieces are within their duelling range.

In reality pieces other than scouts cannot do jump moves. That limits the use of jump moves to the detection of long distance goals. If goals are detected it is possible to evaluate the position by:

- extending the search tree with moves,
- static evaluation of duels.

This preliminary evaluation of goals should be used to determine selection criteria for goals. From the list of candidate goals only a few goals should be selected as the favourite ones that will be given attention in the final choice of a best real move.

In the example on the previous page the jump move K4-K6 covers the distance to the miner at once and eliminates the superfluous node that comes from the move K4-K5. This substantially reduces the search tree and focusses on the real significant moves for both sides.

Generating of jump moves can be realised with shortest path algorithms. It's important to maximise the efficiency of shortest path algorithms in Stratego programs. In Wikipedia references to shortest path algorithms can be found [WP6].

D.2 Selectivity and goal directedness

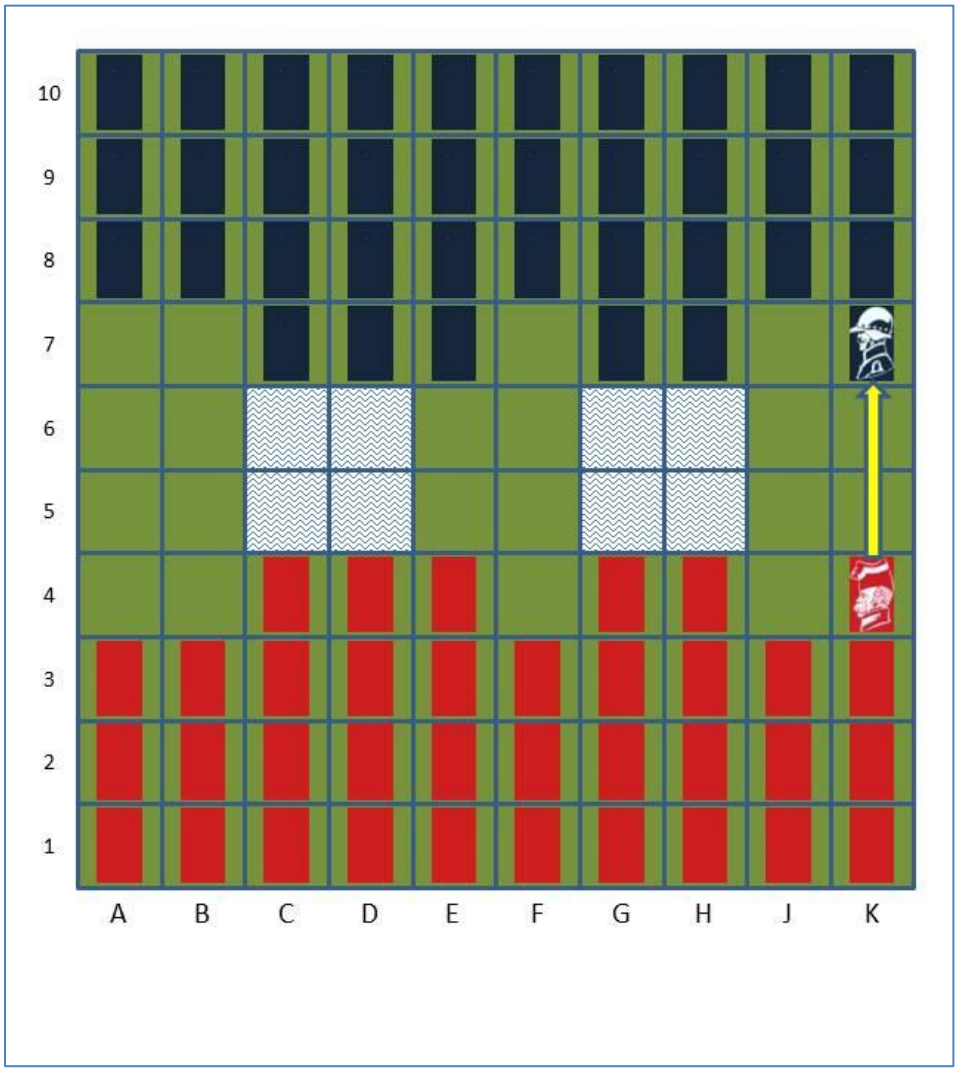
There are good reasons to apply selectivity to local search trees in Stratego. Good players comply with the strategic rule that information about the own ranks should be minimised. If a piece has been moved this exhibits the information that its rank cannot be a bomb or the flag. In order to minimise this kind of information the number of moved pieces should be kept minimal. Therefore any goal should be attained with a minimum of moved pieces. Thus a player only moves an unmoved piece if a good reason for the move exists. If in an area one piece has been moved then there is an obvious preference to move only that piece in that area. A local tree search should therefore mainly consist of moves with one piece directed to one single goal. Only if a second piece is required for the achievement of the current goal then moves of a second piece in the same area may be included in the search tree.

This shows that strategic considerations favour a very drastic selectivity in local search trees. In games like chess there always is a chance that some far hidden and unexpected move may make the difference. This seldom happens in Stratego. Most tactical moves are straightforward attempts to enter or evade a duel and require only moves towards or from a target square by one piece. Therefore a substantial part of goal directed local search trees should consist of levels with two branches:

- One branch for the move of a piece towards or from a target square in the current area
- One branch for not doing a move in the current area.

If a goal has been chosen then goal directedness narrows the width of local search trees to such an extent that the depth can grow to tens of moves within the limits of acceptable processing times. The real effort lies in the detection, evaluation and selection of goals in the search tree. This produces a list of candidate goals from which one emerges as the favourite and that goal is a beacon for the selection of moves in the local search tree.

D.2.1 The implementation of goal directedness in the current example



Red is in turn. Here focus is on the right side. An analysis of goals should produce here a preference for an attack of the red sergeant to the blue miner.

A goal directed local search tree for this position only contains:

- Forward and sideward moves of the red sergeant.
- Sideward moves of the blue miner
  In other positions the tree may contain backward moves for the defender.
- Null moves, when the two squares rule forbids a duel evading move by blue.

In at most 25 moves the blue miner will be captured.

A brute force search tree would require an unacceptable amount of time.
But the goal directed search tree contains a small amount of nodes. Processing time will be drastically reduced to rather acceptable amounts for this kind of positions.

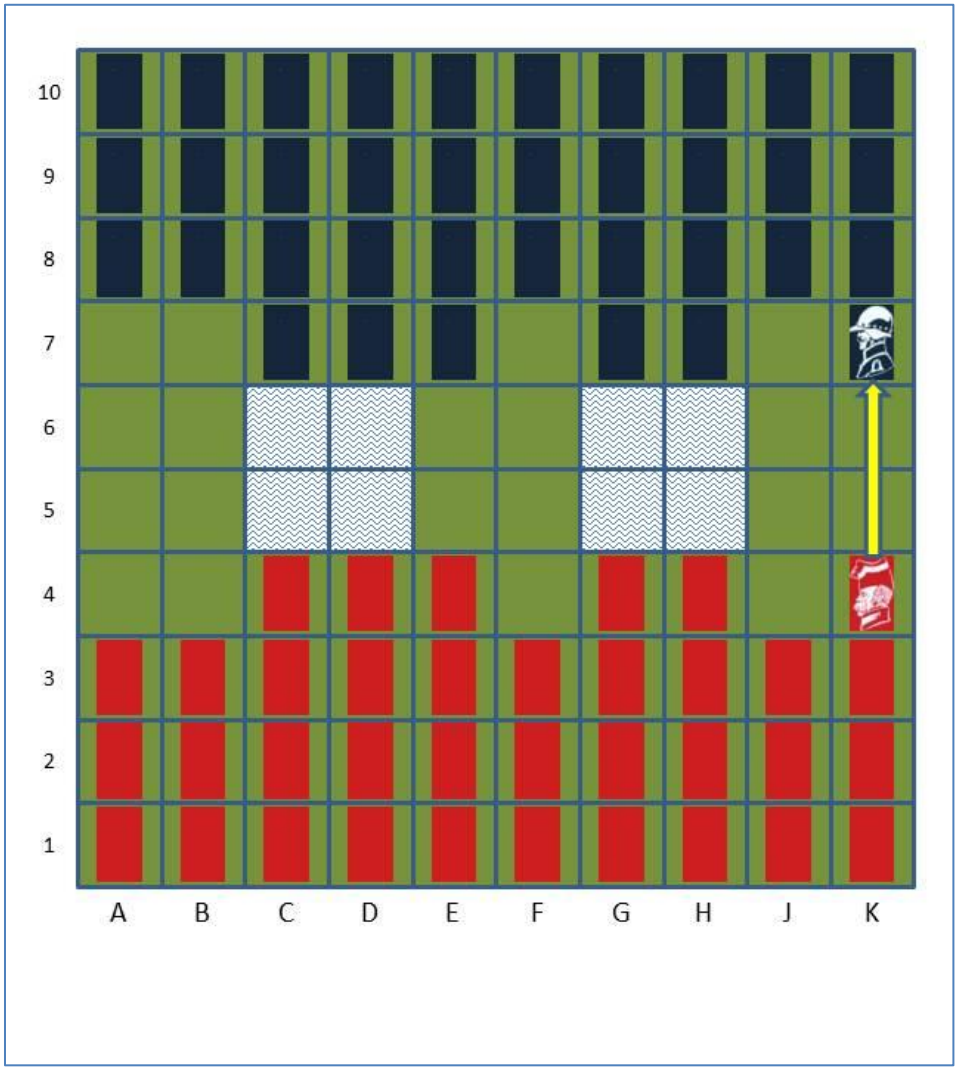
D.3 Look-ahead functions in local search trees

In the preceding appendices some look ahead functions have been mentioned:

- More than two pieces engage into a duel.
- An attacker threatens a piece with a restriction to sideway moves.
- A piece can be encircled.
- A bulldozer attack may be possible.
- Two pieces will be the subject of a simultaneous threat after one of them has been chased.
- Interception of an attacker that tries to attain a target.

A look ahead detection algorithm should be able to determine whether a game position is apt to the application of any of these look ahead functions.

D.3.1 The implementation of a look ahead function in the current example



A look ahead function should be able to detect in this position that the two squares rule is in force. From static data in the position a two-squares rule algorithm should conclude that the blue miner will be captured. So this algorithm should make the current position an end node in the search tree. This example shows why look ahead functions will have a drastic impact on processing times in search trees. The larger the distance between attacker and defender, the larger is the profit of look ahead functions.

Appendix E Version history

Version	Date	Comment
1	August 2012	
2	June 2017	Paragraph 8.3: project activities require more time
3	September 2018	Paragraph 2.6: about computer learning for Stratego