

Stratego Programming Wiki (first draft)

Note.....	1
Scope.....	1
Current state-of-the-art.....	1
Programming objective.....	2
Measurement and analysis.....	2
Board and piece representation.....	3
Search.....	3
Shortest paths.....	5
Setups and Flag detection.....	5
Flag protection.....	8
Trapping pieces and avoiding traps.....	10
Information and risk management.....	10
Assigning relative piece values.....	12
Situational logic.....	13
Table A: Flag Shield Statistics.....	14
Table B: Setup Position Piece Odds.....	16
Table C: Spy Relative Position.....	20
Other resources.....	22

Note

This is a first draft for a wiki to be put up on the StrategoUSA site. It has not yet been reviewed by other authors.

Scope

This wiki is a repository of analysis and discussion on topics related to the creation of computer programs that play the game Stratego.

You can learn more about Stratego here:

<http://en.wikipedia.org/wiki/Stratego>

Current state-of-the-art

As of January 2010 there are three programs that play better than many novice human players, and can win some games against opponents with average abilities. There are: Master of the Flag by Sven Jug, Invincible by Vincent de Boer and Probe by Imer Satz.

You can learn more about these and other Stratego programs, as well as the computer Stratego world championships, here:

http://www.strategousa.org/wiki/index.php/Main_Page

Master of the Flag can be played online here:

<http://jayoogee.dyndns.org/masteroftheflag2/default.html>

A trial version of the program Reveal Your Rank! can be downloaded here:

<http://www.yellowgames.com/>

Probe is available as a free standalone application here:

<http://www.probe.imersatz.com/>

Probe also operates as a bot opponent on the Metaforge online game site here:

<http://www.metaforge.net/olg/webstratego/>

Programming objective

The objective of this wiki is to assist creation of programs capable of playing on a par with human opponents of average skill level.

Discussion: The objective states “human opponents.” Stratego is a game of imperfect information, and as such sits somewhere between chess and poker. A successful Stratego program will be one that is able to exploit human bias and imperfect memory, and at the same time remain impenetrable to human learning. Therefore, the skills required to defeat human opponents are not identical to those necessary to win against other computer programs. In fact, it is entirely possible there are radical uses of misdirection and other ploys that would prove useful against human subjectivity, but not reveal themselves in play against other programs

The objective states “opponents of average skill level.” Stratego is an arcane game with many exceptions and special cases that can easily bog down a programming effort. By the same token, Stratego masters employ sophisticated strategies for piece development and board control that are beyond the capabilities of present programming efforts. This wiki is devoted to improvements in core functionality: search, shortest path, inference, Flag detection, piece traps and so forth. A Stratego program that performs these tasks well would present a strong challenge to average human opponents, even in the absence of attention to many of the fine points of the game, which, in any event, can be added later.

Measurement and analysis

The first three computer Stratego world championships revealed a large gap between the top three programs and the remaining entries. It also suggested the best three programs play about equally against each other. In general, these are the areas in which the weaker programs most need improvement:

- **Play less aggressively.** Most of the weaker programs launched too many attacks against unknown pieces, and eventually lost their power pieces.
- **Manage risk better.** The programs moved too many known pieces adjacent to unknown (and sometimes known!) opponent pieces, and then lost them to attack.

- **Protect the Flag.** The programs advanced pieces on attack without regard to reserving pieces for Flag defense, or knowing how to intercept Flag threats.
- **Conceal the Flag.** The programs put too many pieces into play, which made it easier to locate the Flag. This tendency also puts too many pieces at risk of capture, by clustering moved pieces.
- **Trap vulnerable pieces.** The programs had limited, if any, ability to trap a vulnerable piece using two or more stronger pieces.

Measuring performance against other programs is a manageable, if imprecise, endeavor, and is accomplished through the annual computer Stratego world championships. But there is at present no way to measure programs against human opponents. Perhaps a more useful goal at this time would be to ignore measurement and instead find means to facilitate testing and evaluation against human players. To this end, consider that, over the past two years, Probe bots have played over 50,000 games against nearly 1,000 human opponents on Metaforge. Although some of the games are of questionable quality (players tend to lotto, and then abandon games that are going poorly), this nevertheless represents a huge sample. The Metaforge API supports post-game logging, including full disclosure of the opponent's setups, which allows storage of a game's initial position and moves in a format readable by the Stratego program. Probe can store a position and later replay it, to try out modifications to the AI. Metaforge thus offers an excellent venue to test and evaluate Stratego programs.

Board and piece representation

For simplicity, assume the board is represented by a two-dimensional array, with a surrounding boundary of out of bounds squares.

With respect to each square, the following information is of use:

- Whether the square is vacant, out of bounds, holds one of the mover's pieces or holds one of the opponent's pieces
- For a setup square, the rank of the piece that occupied the square at the start of the game, if known

Search

In a game of perfect information, a search tree can be generated with complete confidence of the position at each node. For these games, the goal of search is to find the objectively best move as seen from a "God's eye" view of the board, and the more combinations searched, the better. This principle does not apply to Stratego. In Stratego the optimal move is based a combination of factors, including what is known on the board, what is believed to be true, what is assumed about the opponent's state of mind and how much risk is appropriate to the position. Search in Stratego usually concerns the effort to maximize the probability of success in future exchanges between presently unknown pieces, rather than the calculation of exchanges between pieces of presently known rank. The game is an exercise in risk management.

At present there is no general agreement on how best to apply search to computer Stratego. Following are general notes on some of the approaches in use by existing programs.

No recursion

One solution is to avoid recursion altogether. Invincible, for instance, does not build a recursive search tree. Instead, all of the look-ahead is performed by a shortest path algorithm and routines that detect specific patterns, such as forced Two-Squares Rule traps. Invincible's success at the first three computer Stratego world championships demonstrates the extent to which other factors besides the exploration of specific move combinations predominate in Stratego.

Full spectrum recursion

Another approach would be to search all possible moves for all possible ranks of each unknown piece, and then weight the results according to inferred piece rank probabilities. But this search tree would be immense. A smaller tree could be generated by limiting branching to attacks, one branch per possible rank, but even this tree could be quite large. However, this approach does seem to mimic human logic, which hedges bets on unknown outcomes.

There is yet another dimension to full spectrum recursion, and that is the opponent's state of mind. So the tree should actually be weighted not only by the probabilities of specific opponent piece ranks, as seen by the computer program, but also the presumed probabilities of the program's pieces, as seen by the opponent.

Guessed piece ranks recursion

Probe makes specific guesses of all moved, unknown opponent pieces, and some unknown unmoved pieces, and then updates the guesses as the game progresses. This permits the use of conventional search tree techniques, but, by itself, produces unsatisfactory results. For example, it may be expected that a piece will defeat each of a series of unknown opponent pieces, but it does not necessarily follow that it will defeat them all. Therefore, with a guessed rank approach, it is necessary to introduce somewhat arbitrary measures such as limiting the number of assumed wins in a move sequence. Nevertheless, guessing does greatly simplify search and helps concentrate attention to specific goals, such as attacking a suspected Spy.

New research in the field

What new research exists for other games that require decision under uncertainty, and that might be applied to Stratego? Here are a few places to look:

<http://www.cs.ualberta.ca/~games/>

<http://www.unimaas.nl/games/>

<http://ticc.uvt.nl/icga/>

Shortest paths

Stratego requires look ahead far too many plies deep to be found by search. For example, in an endgame, a Miner may need to approach a potential Flag location as many as 19 moves distant, plus the intervening opponent moves. And that number grows even larger if it is necessary to advance a stronger piece alongside as support.

The supplement to search is the use of a shortest path algorithm. This calculates the fewest number of moves required to advance a piece safely from point A to point B. There are many useful targets in Stratego: potential Flag locations, Bombs, weaker pieces, pieces that threaten your own Flag, and so forth. Shortest path algorithms produce dramatic results in Stratego: they let pieces know when targets become accessible, which routes to pursue and how to find alternate routes when current routes become blocked.

Finding safe squares

Paths must cross squares that are safe to occupy. Many factors contribute to whether a square may be considered safe. Most obviously, a stronger adjacent known opponent piece renders a square unsafe, if it is presumed the opponent piece will still be at that location when the mover's piece passes through. A potentially stronger unknown adjacent piece may or may not represent an unacceptable risk to the moving piece. Likewise, a succession of potentially stronger adjacent unknown pieces may represent an unacceptable risk; for example, a known Miner may be able to safely pass alongside one unknown opponent piece en route to a Bomb target, but is very unlikely to get past three. Finally, a square may be considered unsafe if there exists a stronger opponent piece that could—and is expected to—intercept the moving piece at that square.

Current algorithms in use

Invincible uses the Dijkstra algorithm.....

Probe calculates paths at each node, a task that occupies about 75% of total evaluation time. For each piece, Probe marks all safe adjacent squares as 1's, then all non-overlapping safe squares adjacent to those as 2's and so forth until all safe squares on the board have been reached. Then it identifies reachable targets, records the distance to the target and recursively traces backwards to find the next move(s) by the piece that approaches the target. Probe does not make assumptions about opponent piece moves. Instead, it observes the board as static and calculates paths based on the current piece positions. However, since Probe also has logic to calculate the value of piece trapping opportunities, it does account in other ways for the potential risk of moving along the shortest path. A better routine would calculate the "shortest-safest path" that accounted for both distance and risk.

Setups and Flag detection

The ability to detect the opponent's Flag plays a surprisingly small role in Stratego. Usually, the Flag is well protected and only becomes accessible late in the game, when its location is apparent through piece attrition. But, all else being equal, it is obviously beneficial to recognize likely Flag locations.

The online game site www.gravon.de has published an archive of games played between players of various skill levels. Imer Satz analyzed over 150,000 setups to count frequencies of the following types of bomb shields:



Corner close shield



Corner far shield



Edge close shield



Back row close shield



Back row far shield



Back row double box shield



Middle close shield

These setups represent 97% of all bombed setups in the archive, of which the back row close shield accounts for 60%. However, fully 56% of the setups did not bomb the Flag at all, and nearly half of those setups had no Bombs adjacent to the Flag. Thus we see that human players generally prefer to place the Flag where it is hardest to identify, rather than attempt to fortify and defend it. There are other interesting patterns, such as a very significant preference for fortifications in the right corner over the left. See Table A for a complete statistical breakdown of setup patterns.

Of course, during a game, bomb fortifications rarely reveal themselves in their entirety. Instead, a player will leave a certain number of adjacent mobile pieces unmoved to conceal the fortification, and leave bluff fortification patterns of unmoved pieces elsewhere (although complete bluff bomb shields with, say, a Sergeant inside are comparatively rare). A good Flag detection algorithm should be able to weight the odds of a particular piece being the Flag (or Bomb or unmoved mobile piece) by considering both the general statistical odds for that square being of a particular rank and the pattern of pieces and open squares surrounding it. Note that nearly all fortifications are diamonds or partial diamonds, which renders pattern detection a fairly simple task.

Flag detection algorithms can be tested against actual data. The Gravon database includes the moves for the games. The accuracy of an algorithm can thus be tested at various stages of the game, as progressively fewer pieces remain unknown and unmoved. Since human players introduce unconscious bias in their choice of setups, and have both limited and imperfect memory of the setups they have faced, this is an area where a computer program could exert an advantage over its mortal opponents.

Flag protection

There are at least three ways to defend the Flag: (1) no defense (2) positional defense and (3) potential threat interception. They are discussed below.

No defense

With no defense, the player leaves the Flag in plain sight, and hopes it gets overlooked. This approach makes more sense with the Flag exposed than fortified. It would be interesting to calculate the success rate of games played with an unprotected Flag and

few, if any, protective moves. In any case, it would seem to be good practice for a Stratego program to play a certain percentage of games with an open, undefended Flag.

Positional defense

Positional defense is the maintenance of a protective wall of mobile pieces to defend the Flag, which is presumably fortified. This is reasonably easy to implement, and is effective against weaker opponents, who do a poor job holding Miners in reserve and penetrating bomb shields. But positional defense, if too obvious, would signal the Flag location in a manner an observant opponent would eventually detect. Also, if the wall becomes broken, it is important to have threat interception logic as a backup. Otherwise, during an endgame on an open board, the program will make no effort to defend the Flag.

There is a special case in which the ability to establish a perimeter defense is decisive. It occurs when a player is in a position to categorically defend his Flag against any attack, *and* still have enough free pieces remaining to win the game. The defense can be as simple as controlling the three lanes with invulnerable pieces, when no opponent piece able to reach the Flag has penetrated to the near side. It is usually easy to determine sufficient strength for the defense, but it can be quite complicated to calculate what is required to win the game. This is probably best regarded as an advanced technique, beyond the scope of current programming efforts, although any decent human player would be able to detect this opportunity and put up the wall.

Threat interception

Probe relies exclusively on threat interception to defend its Flag. Probe looks for pieces that could capture the Flag and will try to intercept the piece. Protection against Miners, and potential Miners, also extends to the Bombs in the fortification. Probe uses its shortest path algorithm to find interception routes.

There are at least two potential downsides to this approach. One is that, lacking positional defense, Probe may not happen to have a piece in a position to intercept a threat when one arises. But in practice, this is quite rare because Probe detects and responds to threats from pieces quite a few moves from the Flag, which at the same time broadens the area of potential interceptors. The other problem is that an overly active defense response might telegraph the location of the Flag. To counteract this Probe calculates how likely is the opponent to have guessed the Flag's location, and also how likely to risk the piece in question by attacking it before deciding whether to intercept the piece.

Confusion and misdirection

Flag defense benefits by maximizing the number of potential Flag locations. Thus, a Lieutenant may be more valuable as an unknown unmoved piece on a location likely to be the Flag or a Bomb than as a piece moved out on the board in pursuit of other objectives. And it will be even more valuable if it either obscures the Flag fortification or contributes to a bluff fortification.

Trapping pieces and avoiding traps

To trap a piece, it is necessary to identify a target, box it in and capture it. This is perhaps the one area of play in which existing Stratego programs are most demonstrably inferior to human players, insofar as these programs frequently overlook—or don't know how to complete—forced trap opportunities.

The first task is to choose a target. If one or more of the potential targets is unknown, the various risk/reward scenarios must be explored. This uncertainty also extends to pieces that may be able to protect the target. Also, clusters of targets must be considered, since it may be preferable to capture two or more weaker pieces than one strong one.

Invincible traps a piece by progressively reducing its range of movement. If a piece can further constrain the target, it will do so. If not, another piece will approach the target until a new constraining move is created, and so on, until the target is captured. The target's free range is calculated as the number of squares it can reach before any stronger opponent piece can reach them. Probe also has an implementation of this idea, as suggested by Vincent de Boer.

Avoidance of traps can be seen as the opposite of initiating them. In the simplest case, a pursued piece should move toward open ground. However, clustering also factors into selection of the preferred escape move. Stratego programs often flee threats in the direction of other vulnerable pieces, with the result that at least one is captured. If piece trapping logic is applied to both sides of the board, the Stratego program should choose the best escape paths for its pieces as represented by poorer opportunities for capture by the opponent.

Information and risk management

The player who does the better job minimizing his own risk, and maximizing his opponent's, will probably win the game. This section describes a variety of means to gain/conceal information and inferences about mobile pieces.

Initial piece position

The initial placement of a piece suggests a great deal about its likely rank. For example, the Spy is about one hundred times more likely to be placed in the front row behind a lake than in the front row along an open column (surprising it's not higher!); further the Spy is extremely likely to be placed in the second row behind a lake. Spy location also usually corresponds to the placement of the General and Colonels, so that as these pieces become exposed, the specific location of the Spy becomes more likely. Table B shows piece rank probabilities for all pieces at all squares, based on analysis of the Gravon game archive. Table C quantifies the extent to which Spy placement corresponds to the placement of the General and Colonels, revealing that nearly half the time the Spy is placed adjacent to the General.

Piece movement

Players often suggest the ranks of pieces by how they move them. For example, most players vigorously protect the identity of their Marshals, and will move away from

potential encounters with weak pieces. So, too, many players have a tendency to make the first move of their Spy within a few moves of the opponent's Marshal being revealed. These and many other common patterns are subject to statistical analysis. It would be a worthwhile project to write a tool that is able to search the Gravon archive in search of telltale patterns that could be used against opponents. The Achilles heel of the (generic) human opponent is detectable patterns in his play of which he remains unaware.

Ignorers and aggressors

An unknown piece that ignores a threat, or creates one, suggests its rank, or rather a range of potential ranks. It's possible the piece is bluffing or wishes to make a scouting attack but, as a general rule, this behavior indicates a piece of superior strength.

Defenders

When an opponent piece that is adjacent to an unknown opponent piece is threatened and holds its ground, or moves away from the threat to become adjacent to an unknown opponent piece, that behavior may suggest a range of potential ranks for the piece that appears to be defending it. If it is threatened and an unknown piece moves adjacent to it, that piece may be defending it.

Flag shield threats

In most cases, a piece that, were it a Miner, could penetrate a bomb shield should be considered as such, intercepted and attacked. The most notable exception would be a piece that had pursued a known piece into the area of the Flag, or a move in the vicinity of the Flag when its location is presumed to be unknown.

Scout range attack threats

If an unknown piece, as the Scout, could capture the Flag or Spy with a range attack, the path of the piece should be blocked (or the Spy should move off line). This consideration should probably only apply to the first opponent moves in the search tree. Otherwise, a host of spurious multiple range move attack scenarios will arise.

Attack opportunities not taken

If, for example, a player moves a known General adjacent to an unknown opponent piece and that piece does not attack, it is nearly certainly not the Marshal. And if instead the known piece is a Colonel, then the opponent piece is probably not the Marshal or General. Probe remembers attacking opportunities passed over and uses this information if it becomes necessary to make an inference about the piece later in the game.

Scouting attacks

The best way to manage risk is to eliminate it. If a piece is acting as though it is the Marshal or General or a Colonel, it is beneficial to attack with a weak piece and find out. A program like Probe that makes specific guesses about opponent pieces can be selective about scouting targets, for example, by attacking a piece that is acting as though it is the Spy.

Bluffing

A Stratego program can neither accept nor ignore all potential bluffs offered against it, or the tendency will soon become obvious. Instead, the program should make a determination based on assessed risk (with perhaps some randomization thrown in). Factors include: the original position of the piece, the piece's move history, the probability of the piece being of a particular rank given what ranks remain unknown and perhaps the odds an opponent would bluff from this position. Bluffing style would be yet another interesting area for research, although there is probably wide variation among players.

Assigning relative piece values

A simple scheme for assigning piece values would be to use constants for most ranks and variables for ranks whose value depends on whether specific other pieces remain on the board. Probe operates this way, with the following values:

Marshal	600	when the opponent's Spy is off the board
	525	when opponent's Spy is on the board
General	450	
Colonel	350	
Major	250	
Captain	150	
Lieutenant	100	
Sergeant	65	
Miner	200	when one Miner remains
	175	when two Miners remain
	150	when three Miners remain
	125	when four Miners remain
	100	when five Miners remain
	50	if the opponent has no remaining Bombs
Scout	90	when one Scout remains or the opponent's Flag is exposed
	60	for normal conditions
Spy	175	when the opponent's Marshal is on the board
	15	when the opponent's Marshal is off the board
Bomb	175	if the Bomb is protecting the Flag
	75	if the Bomb is not protecting the Flag
	25	if the Bomb is isolated on the board

Piece values also depend on whether the piece is known. Probe assesses penalties ranging from ten percent to fifty percent for exposure of its pieces' ranks. Thus, if an unknown Marshal captures a Lieutenant, the loss of value for the Marshal would exceed the value of the captured Lieutenant, making this a bad move.

The position of the piece on the board is also significant, especially as it applies to trades. In general, a player should prefer a trade with a piece that has penetrated to its side, and decline a trade on the opponent's side, and also prefer a trade when ahead on material and decline a trade when behind.

This simplified approach is probably sufficient for Stratego programs at their current level of play, but it's worth noting that expert players recognize that piece values are, in fact, entirely relative. For example, a Captain late in the game may well be more powerful than a Marshal earlier on. Further, the value of a piece depends not only what pieces remain on the board, but which of them are known and where they are located, because this information determines the mobility of the piece being evaluated. At some point, it would be a worthwhile effort to codify all this, but it would require an advanced player to undertake the effort.

Situational logic

The following considerations are not essential to a good Stratego program, but they do occur frequently enough to be worth attention. They are also not particularly difficult to implement.

Desperation attacks

A piece that is certainly (or even most likely) about to get captured should attack an adjacent unknown piece. By the same token, an opponent piece that becomes trapped should be expected to make a desperation attack on a piece it otherwise most likely would not. Thus, it would be unwise for a Stratego program to run, say, a Major into a mass of the program's own unknown, unmoved pieces in the assumption the Major will simply allow itself to be captured.

Critical threats and lost positions

The assumption of appropriate risk depends on the game position. For example, a player with a Flag in imminent danger of capture, or who is hopelessly behind on material, would do well to attack more aggressively.

Lottoing

One might expect a smooth distribution of risk tolerance across players and relatively consistent tolerance by each player across games. But, in fact, there is a specific strategy called lottoing in which the player throws caution to the wind and attacks recklessly. Lottoing will succeed a certain percentage of the time, because all setups in Stratego are vulnerable to attack. Human players lotto frequently against Stratego programs, if only because they can always terminate the game and start a new one. Lottoing can be measured as repeated unexpected attacks by strong pieces, for which the appropriate response is to get out of the way. In other words, an unknown unmoved piece that might normally remain stationary to bluff as a Bomb might choose to avoid a piece that is lottoing.

Table A: Flag Shield Statistics

160951 Gravon game setups

	Pct All	
Bombed Flag	44.4	
Open Flag	55.6	
Total	100.0	
Adjacent Bombs	Pct All	Pct This Shield
Open Flag with no adjacent Bombs	24.8	44.6
Open Flag with Bomb in front	4.4	8.0
Open Flag with Bomb behind	1.0	1.8
Open Flag with Bomb beside	13.8	24.7
Open Flag with Bombs on two sides	10.7	19.3
Open Flag with Bombs on three sides	0.9	1.6
Total	55.6	100.0
All shields: Flag in shield	Pct All	Pct Bombed
Flag left corner, close diagonal shield	5.7	12.9
Flag left corner, far diagonal shield	0.2	0.4
Flag left edge, close Vee shield	0.1	0.2
Flag right corner, close diagonal shield	8.3	18.8
Flag right corner, far diagonal shield	0.3	0.6
Flag right edge, close Vee shield	0.3	0.7
Flag back row, close Vee shield	26.5	59.8
Flag back row, far Vee shield	0.3	0.6
Flag middle row, close Vee shield	0.6	1.4
Double box	0.8	1.8
Total	43.1	97.2
All shields: Flag in shield	Pct All	Pct This Shield
Flag left corner, close diagonal shield	5.7	52.7
Flag left corner, far diagonal shield	0.2	26.2
Flag left edge, close Vee shield	0.1	3.5
Flag right corner, close diagonal shield	8.3	63.0
Flag right corner, far diagonal shield	0.3	38.0
Flag right edge, close Vee shield	0.3	12.9
Flag back row, close Vee shield	26.5	82.1
Flag back row, far Vee shield	0.3	61.5
Flag middle row, close Vee shield	0.6	14.1
Double box	0.8	45.3
Total	43.1	
All shields: Bluff, bombed Flag	Pct All	Pct This Shield
Bluff left corner, close diagonal shield	2.2	20.0

Bluff left corner, far diagonal shield	0.3	38.0
Bluff left edge, close Vee shield	1.1	51.4
Bluff right corner, close diagonal shield	2.0	15.4
Bluff right corner, far diagonal shield	0.2	33.2
Bluff right edge, close Vee shield	1.2	51.8
Bluff back row, close Vee shield	2.6	7.9
Bluff back row, far Vee shield	0.2	33.7
Bluff middle row, close Vee shield	1.9	43.2
Double box	0.8	46.0
Total	12.5	

All shields: Bluff, open Flag	Pct All	Pct This Shield
Bluff left corner, close diagonal shield	3.0	27.3
Bluff left corner, far diagonal shield	0.2	35.8
Bluff left edge, close Vee shield	1.0	45.1
Bluff right corner, close diagonal shield	3.0	23.0
Bluff right corner, far diagonal shield	0.0	0.0
Bluff right edge, close Vee shield	0.8	35.2
Bluff back row, close Vee shield	3.2	10.0
Bluff back row, far Vee shield	0.0	4.8
Bluff middle row, close Vee shield	1.9	42.8
Double box	0.2	8.7
Total	13.3	

Table B: Setup Position Piece Odds

For each rank, the first line represents the setup back row, and the fourth line the front row. Values are the odds of the piece at the location being of the specified rank, where 1.000 is nominal.

Bombed Flag setups

Marshal

0.043	0.012	0.021	0.010	0.019	0.015	0.011	0.010	0.013	0.021
0.278	0.275	0.294	0.179	0.217	0.266	0.155	0.468	0.365	0.132
0.409	0.427	1.177	1.205	0.906	0.692	1.201	1.248	0.466	0.368
0.102	0.137	1.036	1.760	0.160	0.176	1.982	1.288	0.132	0.077

General

0.029	0.018	0.014	0.012	0.016	0.018	0.014	0.012	0.017	0.020
0.098	0.188	0.289	0.127	0.262	0.275	0.163	0.211	0.142	0.092
0.419	0.630	1.025	1.363	0.973	1.241	1.336	1.094	0.648	0.441
0.101	0.223	0.884	1.736	0.152	0.183	1.977	1.016	0.163	0.133

Colonel

0.098	0.092	0.091	0.065	0.102	0.111	0.073	0.078	0.057	0.067
0.246	0.570	0.769	0.637	0.623	0.631	0.712	0.754	0.708	0.285
0.407	0.547	1.098	0.726	0.591	0.569	0.743	1.431	0.719	0.441
0.100	0.128	0.850	0.702	0.102	0.125	0.612	0.864	0.139	0.091

Major

0.110	0.149	0.342	0.261	0.242	0.228	0.358	0.293	0.121	0.148
0.305	0.613	0.561	0.696	0.643	0.693	0.656	0.651	0.512	0.309
0.593	0.595	0.453	0.503	0.645	0.610	0.443	0.429	0.591	0.570
0.258	0.294	0.696	0.533	0.447	0.360	0.542	0.756	0.301	0.242

Captain

0.186	0.208	0.277	0.283	0.339	0.294	0.242	0.303	0.240	0.158
0.320	0.448	0.382	0.409	0.407	0.439	0.375	0.395	0.449	0.278
0.637	0.595	0.395	0.349	0.589	0.683	0.344	0.355	0.576	0.639
0.499	0.658	0.642	0.535	0.782	0.723	0.530	0.570	0.690	0.531

Lieutenant

0.276	0.329	0.342	0.242	0.217	0.215	0.254	0.289	0.329	0.231
0.372	0.443	0.398	0.315	0.338	0.339	0.317	0.388	0.421	0.326
0.511	0.528	0.368	0.358	0.582	0.480	0.382	0.336	0.480	0.493
0.714	0.892	0.429	0.454	0.861	0.948	0.397	0.465	0.956	0.740

Sergeant

0.736	0.454	0.506	0.390	0.373	0.379	0.345	0.488	0.395	0.619
0.493	0.434	0.350	0.286	0.283	0.294	0.271	0.284	0.476	0.447
0.651	0.371	0.325	0.235	0.301	0.320	0.260	0.241	0.357	0.625
0.815	0.795	0.311	0.314	0.680	0.647	0.318	0.312	0.796	0.779

Miner

1.049	0.837	0.875	0.898	0.767	0.826	0.862	0.968	0.795	1.070
0.493	0.445	0.523	0.553	0.421	0.427	0.527	0.489	0.462	0.435
0.157	0.160	0.369	0.375	0.153	0.153	0.366	0.413	0.201	0.189
0.049	0.032	0.371	0.264	0.032	0.030	0.308	0.310	0.058	0.044

Scout

0.411	0.388	0.240	0.244	0.333	0.314	0.256	0.203	0.318	0.366
0.368	0.443	0.323	0.285	0.489	0.465	0.291	0.292	0.431	0.357
0.529	0.592	0.391	0.431	0.599	0.580	0.430	0.377	0.543	0.521
0.765	0.688	0.404	0.409	0.729	0.746	0.401	0.399	0.640	0.764

Spy

0.045	0.042	0.060	0.096	0.114	0.122	0.070	0.071	0.051	0.046
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

0.164	0.299	0.804	1.185	0.634	0.703	1.411	0.988	0.384	0.174
0.069	0.153	1.127	2.292	0.236	0.348	2.380	1.035	0.165	0.143
0.006	0.003	0.378	0.802	0.011	0.007	0.690	0.430	0.005	0.008
Flag									
2.379	0.520	1.138	2.233	1.782	1.422	2.406	1.574	0.411	3.469
0.031	0.004	0.038	0.038	0.020	0.017	0.041	0.055	0.008	0.117
0.003	0.000	0.008	0.011	0.001	0.002	0.010	0.005	0.001	0.007
0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
Bomb									
0.234	0.879	0.753	0.730	0.780	0.846	0.718	0.712	1.068	0.219
0.939	0.490	0.565	0.630	0.553	0.503	0.631	0.588	0.460	1.085
0.341	0.359	0.255	0.140	0.179	0.195	0.133	0.236	0.379	0.343
0.349	0.201	0.175	0.108	0.099	0.111	0.087	0.149	0.182	0.349

Open Flag setups

Marshal									
0.057	0.056	0.041	0.044	0.069	0.100	0.039	0.047	0.062	0.051
0.240	0.393	0.471	0.373	0.608	0.532	0.284	0.358	0.444	0.226
0.773	0.818	1.172	1.390	0.853	0.830	1.440	1.223	0.922	0.927
0.140	0.228	1.231	1.705	0.190	0.336	1.640	1.571	0.205	0.155
General									
0.057	0.057	0.048	0.039	0.050	0.053	0.046	0.045	0.071	0.056
0.236	0.294	0.322	0.218	0.468	0.445	0.236	0.280	0.277	0.264
0.654	0.732	1.339	1.377	1.306	2.071	1.646	1.297	0.904	0.579
0.226	0.165	1.075	1.601	0.219	0.167	1.908	1.105	0.153	0.162
Colonel									
0.147	0.165	0.136	0.143	0.185	0.170	0.153	0.163	0.164	0.118
0.467	0.687	1.004	0.948	0.795	0.923	0.750	1.200	0.852	0.512
0.561	0.559	1.410	1.004	0.750	0.619	0.968	1.571	0.620	0.544
0.127	0.147	0.810	0.699	0.171	0.150	0.646	0.849	0.164	0.192
Major									
0.226	0.378	0.323	0.416	0.396	0.462	0.385	0.425	0.383	0.235
0.510	0.497	0.848	0.784	0.731	0.768	0.989	0.885	0.561	0.574
0.591	0.550	0.527	0.512	0.716	0.690	0.538	0.534	0.627	0.555
0.279	0.434	0.826	0.804	0.518	0.439	0.800	0.790	0.442	0.298
Captain									
0.275	0.353	0.381	0.385	0.430	0.510	0.449	0.446	0.371	0.265
0.436	0.488	0.494	0.505	0.508	0.546	0.510	0.504	0.482	0.410
0.656	0.706	0.437	0.467	0.611	0.659	0.441	0.443	0.634	0.707
0.814	0.755	0.696	0.672	0.886	0.999	0.603	0.603	0.900	0.807
Lieutenant									
0.359	0.413	0.359	0.415	0.329	0.336	0.327	0.349	0.429	0.382
0.452	0.565	0.461	0.376	0.580	0.411	0.373	0.423	0.520	0.430
0.623	0.618	0.472	0.439	0.614	0.519	0.497	0.495	0.553	0.644
1.006	1.065	0.572	0.532	1.171	1.069	0.496	0.543	1.124	0.906
Sergeant									
0.808	0.644	0.695	0.545	0.575	0.595	0.531	0.664	0.575	0.779
0.717	0.641	0.397	0.372	0.396	0.323	0.383	0.417	0.495	0.717
0.648	0.405	0.383	0.299	0.382	0.401	0.259	0.318	0.450	0.638
0.833	0.924	0.372	0.364	0.876	0.778	0.410	0.489	0.868	0.876
Miner									
1.233	1.007	0.923	0.873	0.979	0.911	0.908	0.905	1.123	1.146
0.632	0.624	0.586	0.694	0.571	0.571	0.717	0.646	0.695	0.621
0.288	0.235	0.462	0.481	0.233	0.239	0.466	0.450	0.302	0.290
0.148	0.102	0.506	0.364	0.084	0.091	0.403	0.463	0.094	0.180

Scout

0.539	0.532	0.341	0.328	0.413	0.420	0.292	0.346	0.488	0.519
0.600	0.588	0.336	0.341	0.524	0.580	0.318	0.326	0.615	0.589
0.669	0.776	0.467	0.502	0.775	0.716	0.520	0.464	0.710	0.675
0.907	0.856	0.493	0.503	0.771	0.803	0.490	0.440	0.796	0.880

Spy

0.114	0.129	0.102	0.161	0.200	0.169	0.170	0.114	0.126	0.135
0.258	0.401	1.153	1.937	1.049	1.133	2.003	1.154	0.420	0.356
0.127	0.141	1.200	2.437	0.285	0.322	2.020	1.171	0.171	0.143
0.037	0.023	0.411	0.860	0.033	0.030	0.957	0.536	0.027	0.032

Flag

0.964	0.560	1.822	3.080	1.652	1.628	3.502	1.884	0.603	1.653
0.172	0.070	0.400	0.672	0.139	0.127	0.780	0.510	0.074	0.227
0.100	0.014	0.086	0.100	0.020	0.022	0.153	0.127	0.019	0.200
0.105	0.013	0.131	0.169	0.019	0.018	0.150	0.189	0.023	0.069

Bomb

0.639	0.842	0.985	0.835	0.864	0.815	0.820	0.904	0.814	0.637
0.749	0.605	0.720	0.598	0.536	0.540	0.547	0.613	0.544	0.731
0.533	0.579	0.472	0.360	0.389	0.408	0.360	0.449	0.562	0.473
0.340	0.316	0.378	0.330	0.261	0.302	0.321	0.400	0.297	0.377

All setups**Marshal**

0.100	0.069	0.062	0.055	0.088	0.115	0.049	0.057	0.075	0.072
0.517	0.668	0.766	0.552	0.826	0.798	0.439	0.827	0.808	0.358
1.181	1.245	2.349	2.595	1.759	1.522	2.641	2.471	1.388	1.295
0.242	0.365	2.267	3.465	0.350	0.512	3.623	2.859	0.337	0.232

General

0.085	0.075	0.062	0.050	0.066	0.071	0.061	0.057	0.087	0.076
0.334	0.482	0.610	0.345	0.731	0.719	0.398	0.491	0.419	0.356
1.072	1.362	2.364	2.740	2.279	3.312	2.982	2.390	1.552	1.020
0.327	0.388	1.959	3.337	0.372	0.350	3.885	2.121	0.316	0.295

Colonel

0.245	0.257	0.227	0.208	0.287	0.282	0.226	0.241	0.221	0.184
0.713	1.257	1.773	1.585	1.419	1.554	1.462	1.954	1.560	0.797
0.968	1.107	2.508	1.730	1.341	1.188	1.711	3.003	1.339	0.984
0.227	0.275	1.660	1.402	0.273	0.275	1.258	1.714	0.303	0.283

Major

0.336	0.526	0.665	0.677	0.638	0.690	0.743	0.718	0.504	0.383
0.816	1.110	1.409	1.481	1.374	1.461	1.645	1.536	1.072	0.883
1.184	1.145	0.980	1.015	1.362	1.300	0.981	0.963	1.217	1.126
0.537	0.728	1.522	1.337	0.965	0.799	1.342	1.546	0.744	0.540

Captain

0.461	0.562	0.659	0.669	0.769	0.803	0.691	0.749	0.611	0.423
0.756	0.936	0.876	0.914	0.915	0.985	0.885	0.899	0.932	0.689
1.293	1.301	0.831	0.816	1.200	1.342	0.785	0.798	1.209	1.346
1.313	1.413	1.338	1.207	1.668	1.723	1.134	1.173	1.589	1.337

Lieutenant

0.635	0.741	0.701	0.657	0.546	0.551	0.581	0.638	0.758	0.612
0.824	1.008	0.859	0.692	0.918	0.750	0.690	0.810	0.941	0.756
1.134	1.146	0.840	0.797	1.196	0.999	0.879	0.831	1.033	1.138
1.720	1.957	1.001	0.987	2.033	2.017	0.894	1.007	2.080	1.646

Sergeant

1.544	1.098	1.201	0.935	0.947	0.973	0.876	1.153	0.970	1.398
1.211	1.075	0.747	0.658	0.679	0.617	0.654	0.700	0.971	1.164

1.298	0.776	0.708	0.534	0.683	0.721	0.519	0.559	0.807	1.263
1.648	1.718	0.684	0.679	1.556	1.425	0.728	0.802	1.664	1.655
Miner									
2.282	1.844	1.799	1.772	1.746	1.738	1.770	1.873	1.918	2.216
1.125	1.069	1.109	1.247	0.991	0.998	1.244	1.134	1.156	1.056
0.446	0.395	0.831	0.855	0.386	0.392	0.832	0.863	0.503	0.479
0.197	0.134	0.877	0.628	0.116	0.121	0.711	0.773	0.152	0.225
Scout									
0.950	0.920	0.581	0.573	0.746	0.734	0.549	0.549	0.807	0.885
0.968	1.031	0.660	0.625	1.013	1.045	0.608	0.618	1.046	0.945
1.197	1.368	0.859	0.933	1.374	1.296	0.950	0.841	1.252	1.196
1.672	1.544	0.897	0.912	1.500	1.548	0.891	0.839	1.436	1.643
Spy									
0.160	0.172	0.162	0.257	0.314	0.291	0.240	0.185	0.177	0.182
0.422	0.700	1.957	3.122	1.682	1.836	3.413	2.142	0.804	0.530
0.196	0.294	2.327	4.729	0.521	0.670	4.400	2.206	0.336	0.287
0.043	0.026	0.789	1.662	0.044	0.038	1.646	0.966	0.033	0.040
Flag									
3.343	1.080	2.959	5.314	3.434	3.050	5.908	3.458	1.013	5.123
0.202	0.074	0.437	0.709	0.159	0.145	0.821	0.565	0.082	0.344
0.103	0.014	0.094	0.111	0.021	0.024	0.163	0.132	0.020	0.206
0.106	0.013	0.131	0.169	0.020	0.018	0.150	0.189	0.023	0.070
Bomb									
0.874	1.721	1.738	1.565	1.643	1.661	1.538	1.616	1.883	0.856
1.688	1.095	1.285	1.228	1.089	1.043	1.178	1.201	1.005	1.816
0.875	0.938	0.728	0.500	0.568	0.604	0.493	0.686	0.941	0.816
0.689	0.517	0.553	0.437	0.361	0.414	0.408	0.548	0.479	0.725

Table C: Spy Relative Position

Database: 160951 Gravon game setups

Distance from Spy to General:

1	72197	44.86
2	36702	22.80
3	16503	10.25
4	13260	8.24
5	9214	5.72
6	6050	3.76
7	3762	2.34
8	1896	1.18
9	880	0.55
10	311	0.19
11	147	0.09
12	29	0.02

Distance from Spy to first Colonel:

1	42140	26.18
2	28380	17.63
3	24745	15.37
4	22973	14.27
5	18597	11.55
6	12088	7.51
7	7107	4.42
8	3123	1.94
9	1289	0.80
10	415	0.26
11	81	0.05
12	13	0.01

Distance from Spy to second Colonel:

1	40182	24.97
2	27996	17.39
3	21059	13.08
4	23727	14.74
5	21693	13.48
6	14248	8.85
7	6889	4.28
8	3194	1.98
9	1280	0.80
10	532	0.33
11	133	0.08
12	18	0.01

Distance from Spy to nearest Colonel:

1	74603	46.35
2	43336	26.92
3	25300	15.72
4	11074	6.88
5	3932	2.44
6	1659	1.03
7	575	0.36

8	319	0.20
9	133	0.08
10	18	0.01
11	2	0.00
12	0	0.00

Distance from Spy to nearest General or Colonel:

1	113046	70.24
2	33158	20.60
3	10095	6.27
4	3153	1.96
5	947	0.59
6	328	0.20
7	142	0.09
8	56	0.03
9	21	0.01
10	5	0.00
11	0	0.00
12	0	0.00

Other resources

Invincible. A Stratego Bot. Vincent de Boer. Master of Science thesis:

<http://www.kbs.twi.tudelft.nl/Publications/MSc/2008-deBoer-Msc.html>

Multi-agent Stratego. Mohannad Ismail. Bachelor of Science thesis:

<http://www.kbs.twi.tudelft.nl/Publications/BSc/2004-Ismail-BSc.html>

(Maarten's paper on inferring piece ranks by movement: can't find link)