RISK Board Game - Battle Outcome Analysis

An example of game-theoretic approaches to analyze simple board games and evaluate globally optimal strategies

Revision 3: "Heroes and Fortresses"

by Harris Georgiou, mailto:xgeorgiou@yahoo.com, 2004.

Abstract

In many cases, including some popular board games like the classic RISK, people tend to follow the same basic principles when it comes in formulating a winning strategy. It turns out that these commonly shared principles of play can be predicted and evaluated by using well-defined mathematical approaches that define the core of the so-called Game Theory. This short study presents the basic principles of game-theoretic approaches in game-playing and applies a simple, yet comprehensive, analysis of the dice-battle outcome in RISK.

Revision 3 of this paper contains useful elements on recent variations of the standard RISK game, including the participation of "Hero" and "Fortress" attributes when calculating battle outcome in specific locations and/or special units. It also contains an extension to the initial game-theoretic model of battle analysis, based on a new probabilistic model that is introduced for calculating the probabilities for each outcome in any single battle, using binomial distribution processes. With such a model at hand, it is fairly easy to decide not only "how" but also "when" to go on battle, given the specific forces opposition.

1. Introduction to Game Theory

The mathematical theory of games was first developed as an analytical model for situations of conflict. It was widely adopted for theoretical studies of economics after mathematicians John Von Neumann and economist Oskar Morgenstern published a book called *"Theory of Games and Economic Behavior"* in the early 1940's. Since then, a wide variety of applications emerged, including co-operative players, timing duels, differential equation modeling and games of imperfect information. In 1950, mathematician John Nash formulated a solid proof for the existence of equilibriums in *non-zero sum* games, the significance of which gave him a Nobel Prize in 1994 for his contribution in Economics, and a new boost in theoretical modeling for absurd problems like *The Prisoner's Dilemma* and the *Cuban Missile Crisis* between USSR and USA during October 1962. In fact, it turns out that most competitive environments in real life, like setting a price for a product in a competitive market, are not zero-sum situations since all adversaries compete for higher gain by not necessarily against each other. Nash's theorem ensures that, sooner or later, the adversaries will be forced to settle down in an *equilibrium* where they can benefit the most possible simultaneous gain for all of them, unless of course they decide to cooperate and raise their common gain even more!

Fortunately, popular board games like RISK are far simpler and far less important than any of these situations. Nevertheless, the desire to win and defeat the opponent is always there, hence the need to formulate a winning sequence of moves and strategies towards this goal. Since in all typical board games each adversary gains by the mere loss of the opponent, the situation is clearly defined by a special subclass of theoretical models of game playing, which is called *zero-sum*. This essentially means that the amount of "gain" a player receives is equal to the "loss" of the opponent. In cases where there are more than one opponents, the single player simply has to play against all of them, which is called *strictly non-cooperative mode* of game playing. If the single player is allowed to cooperate with one or more other players, evidently formulating a coalition, playing against a group of other cooperating or non-cooperating opponents, the original game is transformed into a non-cooperative game between coalitions of players, instead of single players against each other. Of course the theoretical implications of cooperation are far too complex to address by simply assigning each player to a coalition, nor addressing the more general problem of modeling a conflicting situation with more than two distinct opposing sides.

However, Game Theory has proven that all zero-sum games are accompanied by a *globally optimal strategy* for both players, which is the ultimately best way to play for each side, regardless of the movements of the other side. Since even chess is a zero-sum game, it follows that there is a globally optimal way to play. Although the white side has the advantage of playing the first move, it can be proved that this globally optimal strategy will probably lead to a *non-loss* situation due to the nature of the Von Neuman formulation. Of course, the immense complexity of the game of chess has not yet allowed us to exploit, or even verify by computer simulations, this extraordinary fact, as even the best chess-playing computer programs still employ extensive simulation of human experience from chess grandmasters and champions, instead of directly implementing some comprehensive game-theoretic approach to the full. Explicit graph searching algorithms like the classic Minimax and A-B are capable of modeling the game up to the next few hundreds of moves from either side of the players, while modern computers can search through millions of moves per second - still, this is only a very limited view of a complete game of 50-60 moves in chess, which is proportionally equal to a few trillions of possible move combinations. Human chess players are still able to barely compete and win against the best computer players, but it seems that in this particular class of games it is only a matter of time until the computers become fast enough and equipped with so much memory that it would be simple impossible to be forced into a situation worst than a stalemate or a draw. This simple observation is evident in the far simpler game of Checkers, were an experienced player can never loose, or can even win by exploiting mistakes made by the other player.

2. What about RISK?

The game of RISK constitutes of two or more players, opposing their forces on the global map with the ultimate goal of conquering all of the continents. Many common variations include sub-goals and special missions, were each player can win by accomplishing these tasks, even if the other players survive or even occupy more territories at the time. Forces are supplemented and transferred according to specific attributes of the map and the rules of the game itself, while various players can be allies or enemies at different phases of the game. Consequently, systematic mathematical modeling of the game at the strategic (global tasks) and operational (force distribution) levels is quite difficult to formulate. However, opposing forces battling over the ground of a specific region are handled by a small set of combating rules, determined by a set of dices for each opposing side. Thus, the problem of modeling battle outcomes in tactical level (combat) is much simpler and fit to formulate via game-theoretic approaches.

Figure 1: The board of RISK is a game of global dominance!

Standard RISK rules for combat state that each battle round can include up to 3 attacking units and up to 2 defending units, all engaged in combat until the outcome is determined by total elimination or surrender by either side. Earlier versions of RISK prohibited any new unit reformation for either side until one of the opponents was eliminated, or retreat and never attack the specific territory during the same game turn. Now, standard game rules on battles are more flexible, treating dice-drawing as a separate battle, thus each player can define the size of its force before each draw. This means that a battle can now include numerous attacking and defending forces, each participating with up to 3 attacking and 2 defending units. The attacker can always choose to continue the battle with a new draw or stop it entirely. The combat continues with the new forces, as they settle after the previous round by the loss of one unit by either players, and the final winner of the complete battle is the player that is left standing by one or more units on that spot. It should be noted that game rules forbid the defender to introduce more opposing units than the attacker, which means that if the attacker uses 1 unit, then the defender must also use no more than 1 unit too. This rather explicit game rule is the root of much of the asymmetry in modeling the dice battle, as it will be evident further on in this study.

Since each player starts with a fixed, not infinite, number of engaging units and cannot alter them until the end of the battle, and since in each round of dice-drawing one of the players is sure to loose one unit, the battle is sure to be completed in a limited number of rounds. Furthermore, since each player's gains and losses are directly linked to the losses and gains of the opponent, the related game is said to be zero-sum. Only two players can engage in combat at a time, so the game contains only two opposing sides, which is the simplest form of zero-sum games.

The number of "moves" available for each player determine the number of *strategies* that this player can introduce into the game. Here, the number of units dictate the number of dices available to each player, when drawing against the opposing force. The player cannot control the outcome of each dice, hence nor the number that will be used to determine the player's support value on each round, but since the highest number is always selected, it is expected that more dices mean higher chances to draw a relatively high number. From this point of view, the attacker, which is granted by the choice of using up to 3 dices, exhibits an advantage over the defender that can only use up to 2 dices. On the other hand, equal draws work in favor of the defender, which means that the attacker has to draw a strictly higher number in order to win.

In order to formulate a robust game-theoretic model for this situation, the possible battle configurations, in terms of numbers of opposing units, has to be distinctively separated and studied as a special case. This means that a simpler game has to be assigned for each combination of units, so that each case can be analyzed and studied separately. For example, when each opponent introduces 1 unit each the game becomes 1x1, when the attacker uses 2 units and the defender 1 unit the game becomes 2x1, etc. The complete list of all possible game configurations, the first number stating the attacker's forces and the second number the defender's forces, can be constructed as follows:

	Def=1	Def=2
Att=1	1x1	*1x2*
Att=2	2x1	2x2
Att=3	3x1	3x2

Table 1: RISK battle configurations

The 1x2 configuration, as stated before, is invalid according to the game rules, thus it should not be accounted for. Furthermore, this fact invalidates the symmetry of the configurations table, making it a special case of 5-cell structure, rather than a normal 3x2 matrix.

3. Battle simulation

For each game formulated above, it is possible to construct a set of enumerating combinatorial equation, calculating exactly how much of all the dice-drawings will favor each player, evidently describing the gain factor for each one of them. However, the fact of using the higher of the numbers drawn by the dices available for each player when using more than a single dice, essentially sorting out the drawn numbers and selecting the maximum, greatly complicates the problem in combinatorial terms. A complete plan of dice draws and the effected selections have to be formulated for all possible combinations, before someone can apply then into calculating the outcome of the battle. Hence, for 1dice (of 6 sides of course!) the possible combinations are 6, for 2 dices the possible combinations are

6*6=36, for 3 dices the possible combinations are 6*6*6=216, and so on. Each of these cases have to measured against the dice combinations of the other player, which means that for a full battle of 3 attacking units against 2 defending units, one has to construct a 216x36 matrix with all the possible outcomes, determine the gain or loss for both players in each case and finally add all of them up to determine which of the players has a higher overall chance of coming out a winner.

Instead of constructing such large matrices by hand and trying to figure out how they can be formulated in a compact form by using combinatorial equations, a simple computer program can simulate the drawing of the available dices in all possible situations, recording the outcomes in lists and processing them later to determine the final outcome. Specifically, each of the 6 available battle configurations is simulated by a nested loop of dice drawing, determining the results of all possible combinations, sorting out the drawn numbers for each player and comparing their best available support values in each case. Since each dice is considered statistically independent from all the other dices, and since all 6 values of each dice are considered equally probable, no special care has to be employed when implementing the related statistical distribution in the simulation loops. In probabilistic terms, each player's strategy is determined by using N statistically independent variables, each exhibiting uniform distribution. Every nested simulation loop essentially constructs the outcome matrix for the related battle configuration, as described above. The 6 battle outcome matrices constitute the result of the complete combined game.

4. The simulation program

The simple program, presented at the end of this report, is a Matlab source that implements all of the six simulation loops described in detail in the previous section. After evaluating each battle outcome matrix, essentially filling each cell with +1 for attacker's win or -1 for defender's win, it sums up the final result for each battle configuration and determines the *game value* for each of them. Since all of the statistical distributions for the dices are considered uniform, there is no special care about using weighting factors when calculating the sum. Positive game value means that the attacker is favored for a win in that specific battle configuration, whereas a negative value means the same for the defender.

Using the game value for each combined battle outcomes, the complete game matrix can be constructed in accordance to the number of units that each player introduces into the battle. The final game matrix calculated by the simulation program is presented here:

		DEF	
		(1)	**(2)**
ATT	**(1)**	-0,1667	
	(2)	+0,1574	-0,1103
	(3)	+0,3194	**+0,0395**

Table 2: Standard game rules battle outcome matrix

5. Battle analysis

Before commenting about the results obtained by the simulation, some notes have to be pointed out first. Although the battle may contain multiple rounds of dice-drawing, it can be considered as a single-stage game since no player can alter its forces before the battle is completed. A player can always retreat from a battle, but this does not alter the nature or characteristics of the initial battle configuration, it simply interrupts it in an intermediate stage. The total gains or losses for each player can be limited this way, saving some of the units if they are stuck with disadvantageous odds and they are about to loose from a stronger force, but the outcome of each specific battle configuration is determined by the 3x2 matrix of outcomes (except for the invalid cell at: A=1, D=2), calculated by the simulation.

Using the typical theorems and propositions of Game Theory for standard zero-sum games, the 3x2 game matrix calculated above dictates that the attacker exhibits the highest advantage when introducing 3 units against 1 defending unit, as it was expected. The advantage towards the attacker remains even when 3 units face 2 defending units or when 2 units face 1 defending unit. This typical result simply proves that the bonus gained by the defender in cases of equal dice draws can only be overcome by always using a larger number of attacking units, hence more dices than the defender. In order to apply normal matrix operations like max and min functions over rows and columns, invalidating the cell at (1,2) simply means that it does not participate in any similar calculation over a set of cells. For example, since the first row now contains only one valid cell, the maximum and minimum values of this row are equal, i.e. the cell at (1,1). Using this formulation, all further procedures can be applied with no problem at all.

5.1 Optimal strategies for RISK battles

Speaking in strict terms of Game Theory, using the Minimax theorem for finite zero-sum games, the attacker's best strategy is the one that dictates using 3 units (maximum value of minimums of rows), securing at least +0,0395 gain, while the defender's best strategy is the one that dictates using 2 units (minimum value of maximums of columns), limiting the expected damage at +0,0395 at most. The cell at (3,2) is the *saddle-point* for this specific game and the **+0,0395** is the *game value* for both players. Given the fact that each player's choices are independent from the other player's strategy, the attacker should always try to secure as much gain as it can, while the defender should always try to limit its damage to the lowest possible level, in every possible situation. This means that the attacker is advised employ the *dominative strategy* of using 3 attacking units in all cases, while the defender is advised employ the dominative strategy of countering every attack with 2 defending units, whenever these forces are available of course. The game value +0,0395 is the expected outcome of each completed battle game, which means that the attacker has an overall advantage of almost +4% to win any battle when using the best strategy of 3 units, and the defender can limit its disadvantage to -4% when using the best opposing strategy of 2 units.

Since the complete battle game has a saddle-point, the *optimal strategies* for each player are *pure*, that is they should always be followed. In fact, applying the rules of *dominance* over the strategies of rows (attacker’s) and columns (defender’s), it is easy to end up with the same resulting saddle-point without even applying the Minimax theorem. However, since the overall battle can be determined through a series of intermediate dice-drawing rounds, each player can use the battle outcome matrix to decide at any given round if it is best to continue the battle or retreat and preserve the remaining units. From this point of view, the matrix dictates that an attacker with larger force than the defender is always has always the advantage, while a defender with equal or larger force than the attacker's can probably hold-off the attack. Furthermore, the matrix demonstrates that an attack with opposing forces ratio of 2:2 is slightly more balanced than with forces of ratio of 1:1. Although experienced players of RISK should probably know these conclusions by fact, it is not easy to determine exactly how "advantageous" or "disadvantageous" a specific situation is by pure empirical evaluation.

The following graph is a contour plot represents the winning positions for each player. The X-axis contains the possible strategies for the defender (1...2) and the Y-axis contains the possible strategies for the attacker (1...3). The attacker's winning positions are contained within the green area and the defender's winning positions are contained within the white area. The polyline just below the blue zone in the middle defines equilibrium states, were there the outcome of the battle is an exact stalemate. The polyline just above the blue zone defines the plane at level +0,0395 exactly, which is the exact game value. The attacker can hope to achieve more than this in case of lucky dice draws and/or by exploiting limited defending forces, whereas the defender hopes exactly the opposite, however in the long run it is expected that battles should end up around the second polyline, at the boundary between the red and the blue areas of the graph.

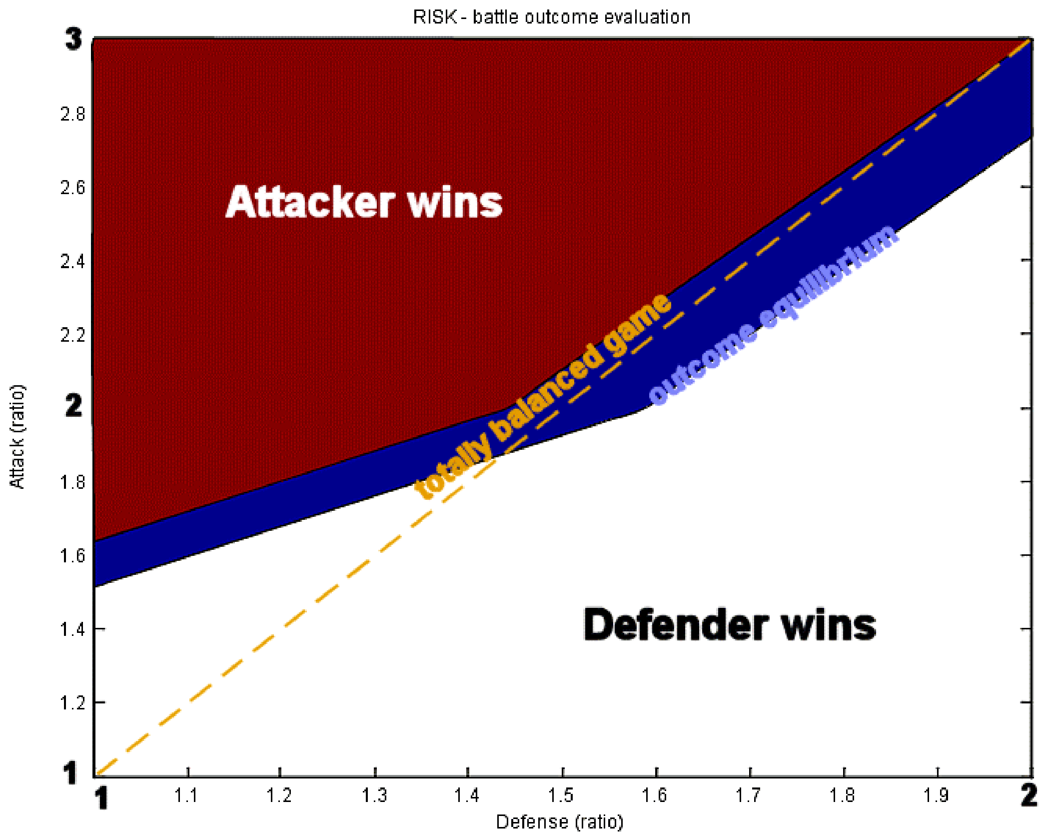

Figure 2: Contour plot for standard RISK battle outcome

The contour plot is generated directly by using the battle outcome matrix, at the last portion of the Matlab source code, using built-in plotting functions that calculate the intersections of the game matrix plane with the flat planes at the given levels, namely the 0 and the game value. However, it is not clear how the zero-level contour polyline, separating the green from the white area, maps directly to the equilibrium curve of the game itself. The 6 matrix values define 6 contour reference points, by which the contour plane is automatically calculated and plotted, hence the automatic generation of the zero-crossing curve. In order to examine this equivalence and calculate the exact points of reference for the equilibrium polyline itself, first we must examine the battle outcome matrix. There are exactly 3 "transitions" in which the game value changes sign, that is crosses the equilibrium curve. These are from (1,1) to (2,1), from (2,2) to (3,2) and from (2,1) to (2,2). These zero-crossings essentially define the three points of the plot that define the polyline, by connecting the two intermediate line sections between them.

5.2 Optimal forces opposition

As the game setup permit only distinct (integer) strategies for both players, i.e. fractional parts are of no concern here, the equilibrium curve here is simply the boundary between winning and losing positions for both players, with no real possibility for true stalemates in the battle. The asymmetry between the attacker (choose more dices) and the defender (wins by default on equal draws) ensures that in every battle there can be only one final winner.

It should be noted that, since the game value and in fact any of the game matrix cells, represents the relative advantage or disadvantage in the related battle configuration, with the attacker favored by positive values and the defender favored by the negative values, it is fairly easy to calculate the exact winning probabilities for each side. A stalemate outcome yields equal a-priori winning probabilities of 50% for each side, whereas a bonus for one side means exactly the same negative effect for the other side, due to the zero-sum nature of the game. As the exact value of this bonus should be exactly one-half of the total difference between these two probabilities, for a game value of gv=+0,0395 the a-priori winning probability for the attacker is: 50%+(gv/2) and for the defender is: 50%– (gv/2):

$$G_{value} = +0{,}0395 \;\; \Rightarrow \;\; P_A=50\%+(G_{value}/2)=51{,}975\% \;\; / \;\; P_D=50\%-(G_{value}/2)=48{,}025\%$$

Using the a-priori probabilities as calculated above, battle stalemate setups can be derived by using a weighted sum of “expected gain” for each side, and solving the equation for a result of zero:

Game value at saddle-point:	G_{value}
Attacker’s winning probability:	$P_A = 50\% + (G_{value}/2)$
Defender’s winning probability:	$P_D = 50\% - (G_{value}/2)$
Attacker’s expected gain:	P_A*N_A
Defender’s expected gain:	P_D*N_D
Cumulative outcome:	$CS = P_A*N_A - P_D*N_D$
On stalemate:	$CS=0 \iff P_A*N_A - P_D*N_D = 0$
Equilibrium forces ratio:	$P_A*N_A - P_D*N_D = 0 \iff P_A*N_A = P_D*N_D \iff$
	$\iff \mathbf{R_e = N_A/N_D = P_D/P_A = (50\% - (G_{value}/2)) / (50\% + (G_{value}/2))}$

Model 1: Opposing forces equilibrium ratio estimation

It is now easy to see how each point of the equilibrium polyline actually maps to opposing forces with specific a ratio, which asymptotically leads to a stalemate in battle. In this case, the game value G_{value}

and the P_A and P_D a-priori winning probabilities that were calculated before, yield an equilibrium ratio of:

$$\mathbf{R_e} = N_A/N_D = P_D/P_A = (48{,}025\% / 51{,}975\%) = \mathbf{0{,}924}$$

The advantageous game value for the attacker has now translated into a slightly lower attacking force size, in a completely "fair" battle. In order not to loose, the attacker should always ensure that it introduces at least 92,4% as much units as the defender, whereas the defender should have at least 8,2% (=1/0,924) more units on the battle area in order to hold off the attack.

5.3 Simulating RISK battles

The zero-sum property of the game, as well as the individual probabilities of gain and loss (+2, +1, -1, -2) are demonstrated in detail on the console output of the simulation program itself:

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=41.667%(15/36)  Def(-1)=58.333%(21/36)
GAME 2x1 PROB:  Att(+1)=57.870%(125/216)  Def(-1)=42.130%(91/216)
GAME 3x1 PROB:  Att(+1)=65.972%(855/1296)  Def(-1)=34.028%(441/1296)
GAME   2x2   PROB:      Att(+2)=11.381%(295/2592)      Att(+1)=38.619%(1001/2592)      Def(-
1)=27.585%(715/2592)  Def(-2)=22.415%(581/2592)
GAME   3x2   PROB:      Att(+2)=18.583%(2890/15552)      Att(+1)=31.417%(4886/15552)      Def(-
1)=35.372%(5501/15552)  Def(-2)=14.628%(2275/15552)

Game value table (rows=attack, cols=defense):

GV =

   -0.1667         0
    0.1574   -0.1103
    0.3194    0.0395

(Att_Hero=0 , Def_Hero=0 , Def_Fort=0)

>>
```

Output 1: Simulation program results for standard battle rules

From the first line of output, regarding the 1x1 battle configuration, the sum of all wins for the attacker yields an overall winning probability exactly the same as the one calculated by the scheme given previously. The game value of –0,1667 divided in half yields a –0,08335 negative bonus over the initial 50% for the attacker, i.e. 41,665%, exactly the same as the value calculated by the simulation (the difference in the last digit is due to rounding in the console output). For better precision, the true value of the a-priori probability can be calculated directly from the number of winning positions versus all dice combinations, which in this case is: 15/36 = 41,6666…%. The same observations can be made for all the other battle configurations, observing the final probabilities for all possible outcomes. It is interesting to see how these probabilities are distributed in the case of +2 and –2 battle outcomes. In all cases, the weighted average of each outcome count, multiplied by the related positive or negative gain (+2, +1, -1, -2), yields the game value for each case.

6. Battle outcome prediction

The analysis of the battle produced useful hints about the optimal strategies and forces opposition for both adversaries, according to the *asymptotic* behavior of the dice-drawing procedure. In Game Theory this may be enough, however in practical situations where there is the need to know, or at least make a good estimation, about the most probable outcome of a specific "stochastic experiment" like drawing a few dices together, one final step has to be made to complete an analytical model for automatic battle planning that can be used by both human and computer players.

Although the analysis of the game shows that the two players should eventually come into a saddle-point, where they both use the maximum of their capabilities in terms of force size, and although it is certain that in this case the outcome of these battles should slightly favor the attacker in the long run, however it is not certain at all whether a specific battle will be won by the attacker or the defender. This fact is of course the result of the stochastic parameter of dice-drawing, where the uncertainty in knowing the drawn numbers introduces uncertainty in knowing the winner altogether. It is rather uncomfortable for the attacker to follow precisely the optimal strategy calculated above, always attacking with an advantageous forces ratio, and still loose some of the battles on the board.

To address this problem, a new element has to be introduced into the player's planning method, specifically a model for the randomness of the dices and the uncertainty they produce. Since the main focus here are not the dices themselves, but rather the outcome of each battle round, the model should address the resulting gain or loss in terms of opposing forces. There are only two possible outcomes for the players in each round, "win" and "loose", that yields +1 or –1 accordingly in the available units in the next round. It is possible for a player to loose more than 1 unit in a battle round if the other player is lucky, but since these situations have already been accounted for when calculating the final game value, it is safe to use the a-priori winning probabilities as the base for a binomial distribution model of "wins" and "losses" for each player. In fact, since the game is zero-sum, calculating the attacker's winning probability distribution automatically yields the defender's model as well.

In order for the attacker to win in a specific battle with specific forces, it is necessary to exhibit at least so many "wins" as the defender's force size, essentially eliminating it. A valid winning position demands at least one surviving unit for the winner, so the available (maximum) number of dice-drawings can be determined directly by the size of the opposing forces, provided the fact that according to the current game value calculation only –1 losses are accounted for.

Modeling the attacking and defending forces into a binomial distribution probability model, the chances of a final win can be calculated analytically by a simple formula:

Attacker's force size:	N_A
Attacker's support probability:	P_A
Defender's force size:	N_D
Defender's support probability:	P_D
Maximum battle rounds:	$Z = N_D+N_A-1$
Winning condition for Attacker:	$\text{Prob}\{ \text{wins}(A) >= N_D \} = 1 - \text{Prob}\{ \text{wins}(A) < N_D \}$
Binomial distribution function:	$\text{Prob}\{X\} = P(x;P,N) = \text{Comb}(N,x) * P^x * (1-P)^{N-x}$

Model 2: Winning probabilities estimation

This model strictly determines the probabilities related to the attacker's "x" wins in a total of "N" battle rounds. Substituting N with Z, P with P_A and x with N_D and using the cumulative binomial distribution formula for summing up all values up to (but not containing) ND wins, it is now possible to calculate exactly the probability of the attacker's force eliminating the defender's force and still keeping *at least* one unit alive at the end. The exact value of this probability depends solely on the battle configuration, which in this case is assumed to be the optimal strategy for both players using the maximum allowable force sizes in every round, the a-priori winning probabilities for the players (i.e. the game value), and the exact size for each of the opposing force. The table below presents a standard probabilities matrix for this situation, showing force sizes up to 15 units:

A\D	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	71,65%	53,70%	38,15%	26,02%	17,20%	11,09%	7,00%	4,35%	2,66%	1,61%	0,96%	0,57%	0,34%	0,20%
4	83,62%	69,25%	54,31%	40,73%	29,43%	20,62%	14,08%	9,40%	6,17%	3,98%	2,53%	1,59%	0,99%	0,61%
5	90,81%	80,45%	67,90%	54,85%	42,64%	32,07%	23,43%	16,69%	11,64%	7,96%	5,35%	3,54%	2,32%	1,49%
6	94,95%	87,99%	78,34%	67,06%	55,33%	44,16%	34,20%	25,79%	18,99%	13,69%	9,69%	6,74%	4,61%	3,12%
7	97,27%	92,81%	85,86%	76,83%	66,51%	55,77%	45,41%	35,99%	27,83%	21,04%	15,59%	11,34%	8,11%	5,71%
8	98,54%	95,79%	91,02%	84,21%	75,71%	66,13%	56,18%	46,49%	37,53%	29,61%	22,88%	17,33%	12,90%	9,45%
9	99,23%	97,58%	94,43%	89,52%	82,89%	74,84%	65,88%	56,57%	47,42%	38,87%	31,19%	24,53%	18,95%	14,39%
10	99,60%	98,63%	96,61%	93,20%	88,25%	81,81%	74,16%	65,71%	56,93%	48,25%	40,06%	32,60%	26,05%	20,45%
11	99,79%	99,23%	97,97%	95,68%	92,11%	87,16%	80,92%	73,61%	65,60%	57,27%	49,00%	41,13%	33,88%	27,43%
12	99,89%	99,57%	98,80%	97,31%	94,81%	91,14%	86,23%	80,17%	73,17%	65,54%	57,60%	49,69%	42,10%	35,05%
13	99,94%	99,77%	99,30%	98,34%	96,65%	94,00%	90,27%	85,42%	79,54%	72,81%	65,51%	57,91%	50,32%	42,99%
14	99,97%	99,87%	99,60%	99,00%	97,87%	96,01%	93,25%	89,49%	84,71%	79,00%	72,52%	65,50%	58,21%	50,90%
15	99,99%	99,93%	99,77%	99,40%	98,66%	97,39%	95,40%	92,56%	88,79%	84,09%	78,53%	72,27%	65,52%	58,50%

Table 3: Standard game rules winning probabilities matrix

Given a matrix like the ones presented above, it is fairly easy for a non-expert player to exhibit optimal battle behavior, making sure that enough forces are allocated into an attack or ensuring a hard opposition in the defense. Models like these can also be used for a snapshot estimation of a player's status, essentially calculating how dangerous an attack can be or how probable is for the opponent to attack a specific region. For every board game like RISK, where the conquer and control of areas in a "map" are a matter of outmost importance for winning the overall game, solving the game at the tactical (lower) levels effectively transforms the task of strategic planning into a logistics problem for optimal forces allocation.

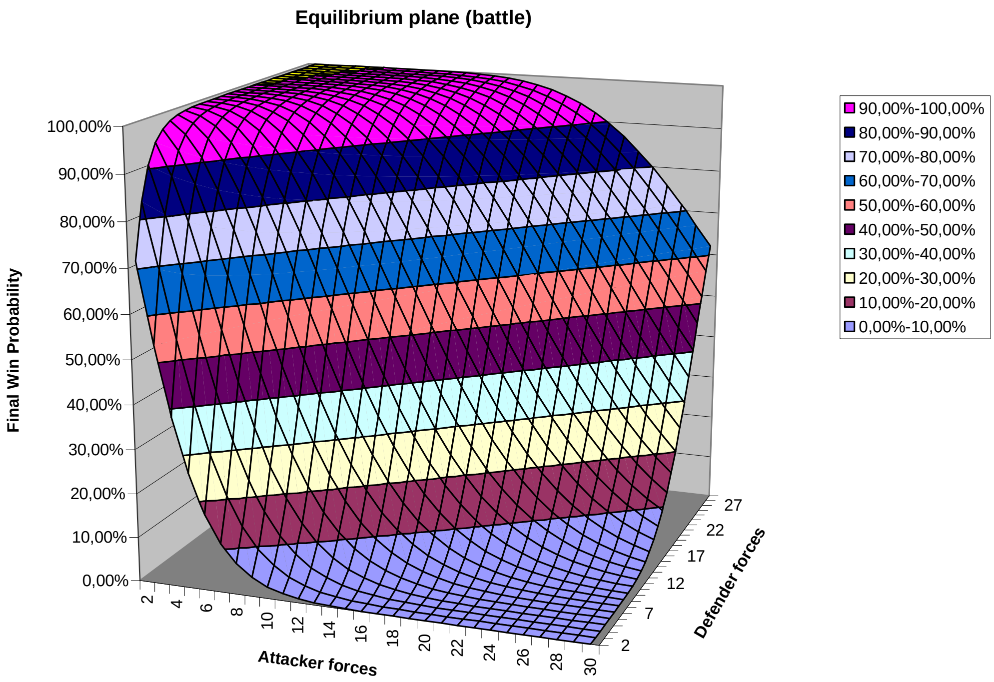

Figure 3: Standard game rules winning probabilities 3D plot

7. Heroes and Fortresses

In some recent variations of the standard RISK game, some new and interesting properties have been introduced into the battles for conquering new areas on the map. Special characters, the "hero" pawns can accompany an attacking or defending force, effectively yielding a special bonus during the dice-drawing phase of the battles. Same this applies for a "fortress" in the area that is to be defended, yielding an added bonus for the defending force.

The "hero" pawn moves and behaves as any other unit of the player, with some added tasks of completing independent sub-missions and claim the accompanied rewards, which however are now directly related with the outcome of an ongoing battle after all. It does not constitute a fighting unit by itself, but if a "hero" is used along with a fighting force of normal units, it adds a bonus +1 to the highest (only) number drawn by the dices used in a round of battle. Thus, if an attacker draws three dices from which the highest number is 5, then this number is calculated as if it were a 6. This bonus applied even if the highest number is already 6, yielding a 7. It the main force wins, the "hero" pawn moves along with it for the current game turn, whereas if it looses, the "hero" unit looses also and it is temporarily removed from the game board.

The "fortress" property applies only for the defender, even if the attacking force has also a "fort" in the area from which it launches the attack, obviously because a static fort cannot be used directly for attacking a neighboring area. The defender, if a "fort" is available in the defending area, adds a bonus of +1 to the highest drawn dice number, even if a similar bonus is already available through a "hero" pawn. This effectively means that a strong defense that exhibits the bonuses from a "fort" and a "hero" in the area can add +2 to the highest drawn dice number, making the battle very hard for the attacker to win unless a strong force is allocated to it.

These two slight variations of the standard battle rules may seem trivial, but in effect they produce serious deviations from the properties and tactical characteristics of the standard battle, as it will be proved in the following sections.

7.1 Attacking with a "hero", defending with standard force

Using the models and formulations already presented in detail for the standard battle rules, it is fairly easy to implement these new rules into the battle outcome estimators of the simulation program, run the new simulation and observe the results.

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=58.333%(21/36)  Def(-1)=41.667%(15/36)
GAME 2x1 PROB:  Att(+1)=74.537%(161/216)  Def(-1)=25.463%(55/216)
GAME 3x1 PROB:  Att(+1)=82.639%(1071/1296)  Def(-1)=17.361%(225/1296)
GAME 2x2 PROB:  Att(+2)=15.625%(405/2592)  Att(+1)=34.375%(891/2592)  Def(-
1)=34.375%(891/2592)  Def(-2)=15.625%(405/2592)
GAME 3x2 PROB:  Att(+2)=25.527%(3970/15552)  Att(+1)=24.473%(3806/15552)  Def(-
1)=40.811%(6347/15552)  Def(-2)= 9.189%(1429/15552)

Game value table (rows=attack, cols=defense):

GV =

    0.1667         0
    0.4907         0
    0.6528    0.1634

(Att_Hero=1 , Def_Hero=0 , Def_Fort=0)

>>
```

Output 2: Simulation program results for added bonus battle rules (ATT=+1 / DEF=0)

Since no other change has been made in the standard battle rules, there are still up to 3 dices available for the attacker and up to 2 dices available for the defender, thus the basic game configurations are still the same. The difference here is that, due to the presence of the "hero" pawn for the attacker, a bonus +1 is added to the highest dice number drawn in every round of the battle.

		DEF (0)	
		(1)	**(2)**
ATT (+1)	**(1)**	+0,1667	
	(2)	+0,4907	0,0000
	(3)	+0,6528	+0,1634

Table 4: Added bonus game rules battle outcome matrix (ATT=+1 / DEF=0)

Using the same game-theoretic analysis for the game matrix produced by the simulation program, it is fairly easy to try and locate a saddle-point, if it exists. In fact, this saddle-point could be in the cell at (3,2), as it were for the standard battle rules, if only there wasn't that little difference of +0,0033 towards the cell at (1,1). If the Minimax theorem is used for the attacker, a preferred strategy of 1 unit force emerges at cell (1,1), whereas if it is used for the defender a force of 2 units is proved to be the best choice. Strictly speaking, in this case the game-theoretic models become complex and involve minimization of linear equations in order to produce a *mixed* strategy for each player, one that defines the optimal ratio of mixing some or all of the available strategies and using them in turns, using a random device that employs the same outcome distribution (like a "weighted" dice) to choose exactly which one to use in each round of the battle. This situation is rather uncomfortable when using normal dices and simple gameplay, however optimal strategy in this case should involve mixing a few of the available pure strategies at hand, thus producing a stochastic way for both players of deciding the exact force sizes that are to be used in the battle.

Fortunately, the battle rules state specifically that the attacker has the initiative of deciding first the size of the attacking force, before the defender does the same for the defending force. Furthermore, the defending force is sure to be of equal or less in size than the attacking force. These two restrictions are enough to clear out any ambiguities on describing the preferred strategies for the two players. Since the attacker has the initiative of stating first the size of its force, the obvious choice is to use only 1 attacking unit that exploits the +1 bonus in dice-drawing due to the presence of the "hero" pawn. This choice effectively restricts the defender of using only 1 unit also, but not having the advantage of the +1 bonus as the attacker has. This situation is the emerging saddle-point, produced by the specific game rule restrictions and not by the game-theoretic properties of the game matrix. In any case, the attacker can successfully lead the game into a situation where a positive game value can be reached and hence secure an advantageous position in the upcoming battle. Its is interesting to observe from the output of the simulation program how the 1:1 confrontation is slightly more promising for the attacker in relation with the 3:2 confrontation, even though in the second one the attacker has more than 25% chance of gaining a +2 result from a every battle round. It is also interesting to see how the "hero" bonus yields to an exactly "fair" battle in case of 2:2 confrontations.

As before, it is fairly easy to use the emerged game value to calculate the a-priori winning probabilities for both the attacker and the defender, as well as the equilibrium forces ratio:

$$G_{value} = +0{,}1667 \;\Rightarrow\; P_A=50\%+(G_{value}/2)=58{,}335\% \;\;/\;\; P_D=50\%-(G_{value}/2)=41{,}665\%$$

$$\mathbf{R_e} = N_A/N_D = P_D/P_A = (41{,}665\% \,/\, 58{,}335\%) = \mathbf{0{,}7142}$$

Even if the change in the standard battle rules effect only in a +1 dice bonus for the attacker due to the "hero" pawn, the advantage is a very strong one since it allows launching successful attacks even with less than 72% available units in relation to the defending force. The presence of a "hero" pawn is proved to secure almost 21% less units for the attacker, in contrast to the standard battle situation.

The following matrix demonstrates the differences on the probable battle outcomes, in contrast to the same matrix calculated over the standard battle rules:

A\D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	58,34%	34,03%	19,85%	11,58%	6,76%	3,94%	2,30%	1,34%	0,78%	0,46%	0,27%	0,16%	0,09%	0,05%	0,03%
2	82,64%	62,39%	44,66%	30,88%	20,83%	13,79%	9,00%	5,81%	3,72%	2,36%	1,49%	0,93%	0,58%	0,36%	0,22%
3	92,77%	80,11%	65,34%	50,98%	38,42%	28,16%	20,18%	14,19%	9,83%	6,71%	4,54%	3,03%	2,01%	1,32%	0,87%
4	96,99%	89,95%	79,70%	67,73%	55,52%	44,12%	34,14%	25,83%	19,16%	13,98%	10,04%	7,12%	4,99%	3,46%	2,38%
5	98,74%	95,08%	88,67%	79,95%	69,77%	59,08%	48,69%	39,17%	30,83%	23,81%	18,07%	13,51%	9,96%	7,26%	5,22%
6	99,48%	97,65%	93,91%	88,09%	80,46%	71,55%	62,03%	52,50%	43,47%	35,28%	28,11%	22,03%	17,00%	12,94%	9,73%
7	99,78%	98,89%	96,82%	93,18%	87,88%	81,08%	73,14%	64,54%	55,76%	47,23%	39,26%	32,08%	25,80%	20,44%	15,98%
8	99,91%	99,49%	98,37%	96,21%	92,74%	87,88%	81,74%	74,57%	66,74%	58,61%	50,55%	42,85%	35,75%	29,37%	23,79%
9	99,96%	99,76%	99,18%	97,94%	95,78%	92,49%	88,01%	82,41%	75,88%	68,68%	61,13%	53,51%	46,11%	39,14%	32,74%
10	99,98%	99,89%	99,60%	98,91%	97,60%	95,47%	92,36%	88,22%	83,08%	77,08%	70,43%	63,38%	56,19%	49,08%	42,28%
11	99,99%	99,95%	99,80%	99,43%	98,67%	97,34%	95,26%	92,33%	88,47%	83,73%	78,19%	72,02%	65,42%	58,62%	51,81%
12	100,00%	99,98%	99,91%	99,71%	99,28%	98,47%	97,13%	95,13%	92,36%	88,76%	84,36%	79,22%	73,47%	67,28%	60,83%
13	100,00%	99,99%	99,95%	99,85%	99,61%	99,13%	98,30%	96,98%	95,05%	92,43%	89,07%	84,96%	80,17%	74,80%	68,98%
14	100,00%	100,00%	99,98%	99,93%	99,79%	99,52%	99,01%	98,17%	96,87%	95,02%	92,54%	89,38%	85,55%	81,07%	76,03%
15	100,00%	100,00%	99,99%	99,96%	99,89%	99,74%	99,44%	98,91%	98,06%	96,79%	95,02%	92,67%	89,70%	86,11%	81,91%

Table 5: Added bonus game rules winning probabilities matrix (ATT=+1 / DEF=0)

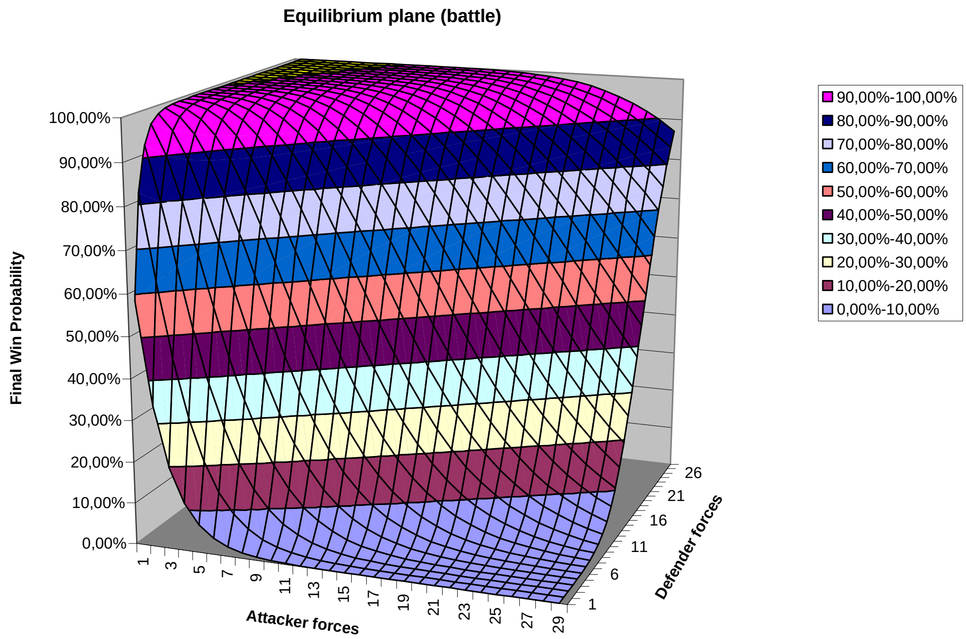

Figure 4: Added bonus game rules winning probabilities 3D plot (ATT=+1 / DEF=0)

7.2 Attacking with a "hero", defending with a "hero" OR a "fortress"

As explained before, the presence of a "hero" pawn gives the player a +1 bonus during the dice-drawing phase of the battle. The same thing applies for the "fortress" bonus, hence means the study of these two situations as individual (not combined) configurations can be treated as one and the same. On the other hand, two forces that each one exhibits a +1 dice bonus can be treated as a standard-rules battle situation, as it can be observed by the output of the simulation program:

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=41.667%(15/36)  Def(-1)=58.333%(21/36)
GAME 2x1 PROB:  Att(+1)=57.870%(125/216)  Def(-1)=42.130%(91/216)
GAME 3x1 PROB:  Att(+1)=65.972%(855/1296)  Def(-1)=34.028%(441/1296)
GAME 2x2 PROB:  Att(+2)=11.381%(295/2592)  Att(+1)=38.619%(1001/2592)  Def(-
1)=27.585%(715/2592)  Def(-2)=22.415%(581/2592)
GAME 3x2 PROB:  Att(+2)=18.583%(2890/15552)  Att(+1)=31.417%(4886/15552)  Def(-
1)=35.372%(5501/15552)  Def(-2)=14.628%(2275/15552)

Game value table (rows=attack, cols=defense):

GV =

   -0.1667         0
    0.1574   -0.1103
    0.3194    0.0395

(Att_Hero=1 , Def_Hero=1 , Def_Fort=0)

>>
```

Output 3: Simulation program results for added bonus battle rules (ATT=+1 / DEF=+1)

It is interesting to see how an attacker's bonus is countered by an equal defender's bonus, with no chance whatsoever to the game itself. Since these bonuses are applied in exactly the same manner in the same situations during the dice-drawing process, the outcome is exactly the same. This fact coincides with the game-theoretic properties of the battle outcome matrix, where the adding of a constant value in exactly the same count of positive and negative battle outcomes has no effect to the final result.

		DEF (+1)	
		(1)	**(2)**
ATT (+1)	**(1)**	-0,1667	
	(2)	+0,1574	+0,1103
	(3)	+0,3194	+0,0395

Table 6: Added bonus game rules battle outcome matrix (ATT=+1 / DEF=+1)

7.3 Attacking with a "hero", defending with a "hero" AND a "fortress"

In this situation, the defender exhibits the added bonus of a "hero" and a "fortress" together (+2), versus only a "hero" bonus (+1) for the attacker. Using the models and formulations already presented in detail for the standard battle rules, the simulation program calculates the new battle outcomes.

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=27.778%(10/36)  Def(-1)=72.222%(26/36)
GAME 2x1 PROB:  Att(+1)=41.667%(90/216)  Def(-1)=58.333%(126/216)
GAME 3x1 PROB:  Att(+1)=49.383%(640/1296)  Def(-1)=50.617%(656/1296)
GAME 2x2 PROB:  Att(+2)= 7.330%(190/2592)  Att(+1)=42.670%(1106/2592)  Def(-
1)=23.341%(605/2592)  Def(-2)=26.659%(691/2592)
GAME 3x2 PROB:  Att(+2)=11.960%(1860/15552)  Att(+1)=38.040%(5916/15552)  Def(-
1)=32.382%(5036/15552)  Def(-2)=17.618%(2740/15552)

Game value table (rows=attack, cols=defense):

GV =

   -0.4444         0
   -0.1667   -0.1933
   -0.0123   -0.0566

(Att_Hero=1 , Def_Hero=1 , Def_Fort=1)

>>
```

Output 4: Simulation program results for added bonus battle rules (ATT=+1 / DEF=+2)

It is interesting to see that a greater dice bonus for the defender produces a negative-valued game matrix, as it was the case with the positive-valued game matrix for the attacker when only an attacking "hero" pawn was present.

		DEF (+1)	
		(1)	**(2)**
ATT (0)	**(1)**	-0,4444	
	(2)	-0,1667	-0,1933
	(3)	-0,0123	-0,0566

Table 7: Added bonus game rules battle outcome matrix (ATT=+1 / DEF=+2)

Using the same game-theoretic analysis for the game matrix produced by the simulation program, it is fairly easy to locate a saddle-point at the cell (3,2), yielding a game value of –0,0566. It is evident that the current situation is exactly the same as the standard-rule battles, only now the defender has the advantage.

As before, it is fairly easy to use the game value to calculate the a-priori winning probabilities for both the attacker and the defender, as well as the equilibrium forces ratio:

$$G_{value} = -0{,}0566 \;\Rightarrow\; P_A = 50\% + (G_{value}/2) = 47{,}170\% \;\;/\;\; P_D = 50\% - (G_{value}/2) = 52{,}830\%$$

$$\mathbf{R_e} = N_A/N_D = P_D/P_A = (52{,}830\% \,/\, 47{,}170\%) = \mathbf{1{,}12}$$

In contrast to the case of a single +1 dice bonus for the attacker, here the added bonus for the defender yields a higher size for the attacking force in relation to the defending force, in order to win the battle. The attacker is now in a disadvantageous situation, because the “hero” bonus is overrun by the double bonus for the defender’s side.

The following matrix demonstrates the differences on the probable battle outcomes, in contrast to the same matrix calculated over the standard battle rules:

A\D	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	64,39%	44,71%	29,23%	18,28%	11,05%	6,50%	3,74%	2,12%	1,18%	0,65%	0,35%	0,19%	0,10%	0,05%
4	77,51%	60,18%	43,83%	30,33%	20,14%	12,94%	8,08%	4,93%	2,95%	1,73%	1,00%	0,57%	0,32%	0,18%
5	86,18%	72,44%	57,33%	43,07%	30,96%	21,44%	14,38%	9,39%	5,99%	3,74%	2,29%	1,39%	0,83%	0,49%
6	91,67%	81,51%	68,74%	55,17%	42,38%	31,32%	22,37%	15,51%	10,48%	6,92%	4,48%	2,84%	1,78%	1,09%
7	95,06%	87,90%	77,78%	65,84%	53,44%	41,75%	31,51%	23,06%	16,41%	11,40%	7,74%	5,15%	3,37%	2,17%
8	97,10%	92,24%	84,60%	74,69%	63,46%	51,99%	41,17%	31,60%	23,58%	17,14%	12,18%	8,47%	5,77%	3,87%
9	98,32%	95,11%	89,56%	81,70%	72,07%	61,46%	50,74%	40,63%	31,62%	23,97%	17,74%	12,84%	9,11%	6,34%
10	99,03%	96,96%	93,05%	87,05%	79,14%	69,80%	59,73%	49,64%	40,12%	31,59%	24,27%	18,23%	13,41%	9,68%
11	99,45%	98,13%	95,45%	91,01%	84,74%	76,85%	67,80%	58,21%	48,65%	39,64%	31,52%	24,50%	18,64%	13,91%
12	99,69%	98,86%	97,06%	93,86%	89,04%	82,60%	74,78%	66,03%	56,85%	47,76%	39,18%	31,43%	24,67%	18,98%
13	99,82%	99,32%	98,12%	95,87%	92,27%	87,16%	80,62%	72,91%	64,43%	55,62%	46,93%	38,74%	31,31%	24,80%
14	99,90%	99,59%	98,82%	97,26%	94,62%	90,68%	85,37%	78,79%	71,20%	62,97%	54,50%	46,17%	38,32%	31,18%
15	99,94%	99,76%	99,26%	98,20%	96,31%	93,34%	89,13%	83,66%	77,08%	69,62%	61,63%	53,47%	45,46%	37,92%

Table 8: Added bonus game rules winning probabilities matrix (ATT=+1 / DEF=+2)

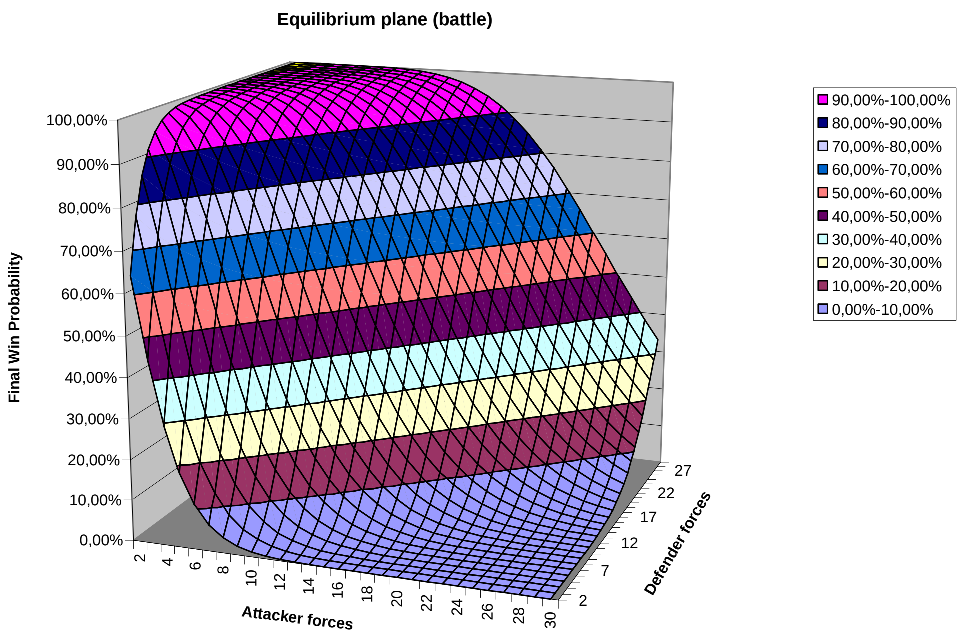

Figure 5: Added bonus game rules winning probabilities 3D plot (ATT=+1 / DEF=+2)

7.4 Attacking with a standard force, defending with a "hero" OR a "fortress"

In this situation, the defender exhibits the +1 bonus that may come from a "hero" or a "fortress", versus no bonus for the attacker. Using the models and formulations already presented in detail for the standard battle rules, the simulation program calculates the new battle outcomes.

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=27.778%(10/36)  Def(-1)=72.222%(26/36)
GAME 2x1 PROB:  Att(+1)=41.667%(90/216)  Def(-1)=58.333%(126/216)
GAME 3x1 PROB:  Att(+1)=49.383%(640/1296)  Def(-1)=50.617%(656/1296)
GAME 2x2 PROB:  Att(+2)= 7.330%(190/2592)  Att(+1)=42.670%(1106/2592)  Def(-
1)=23.341%(605/2592)  Def(-2)=26.659%(691/2592)
GAME 3x2 PROB:  Att(+2)=11.960%(1860/15552)  Att(+1)=38.040%(5916/15552)  Def(-
1)=32.382%(5036/15552)  Def(-2)=17.618%(2740/15552)

Game value table (rows=attack, cols=defense):

GV =

   -0.4444         0
   -0.1667   -0.1933
   -0.0123   -0.0566

(Att_Hero=0 , Def_Hero=1 , Def_Fort=0)

>>
```

Output 5: Simulation program results for added bonus battle rules (ATT=0 / DEF=+1)

As with the situation of countering equal bonuses, it is evident that exactly the same game matrix is produced when using bonuses of 0 for the attacker and +1 for the defender, or +1 for the attacker and +2 for the defender. Consequently, the same saddle-point and game value stand here as well.

		DEF (+1)	
		(1)	**(2)**
ATT (0)	**(1)**	-0,4444	
	(2)	-0,1667	-0,1933
	(3)	-0,0123	-0,0566

Table 9: Added bonus game rules battle outcome matrix (ATT=0 / DEF=+1)

7.5 Attacking with a standard force, defending with a "hero" AND a "fortress"

Obviously, this last configuration is most disadvantageous for the attacker. The defender exhibits the added bonus of a "hero" and a "fortress" together (+2), versus no bonus for the attacker. Using the models and formulations already presented in detail for the standard battle rules, the simulation program calculates the new battle outcomes.

```
>> risk3
RISK board game - evaluation of outcomes, version 1.2
Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com

calculating...

GAME 1x1 PROB:  Att(+1)=16.667%(6/36)  Def(-1)=83.333%(30/36)
GAME 2x1 PROB:  Att(+1)=26.852%(58/216)  Def(-1)=73.148%(158/216)
GAME 3x1 PROB:  Att(+1)=33.333%(432/1296)  Def(-1)=66.667%(864/1296)
GAME 2x2 PROB:  Att(+2)= 3.935%(102/2592)  Att(+1)=46.065%(1194/2592)  Def(-
1)=21.026%(545/2592)  Def(-2)=28.974%(751/2592)
GAME 3x2 PROB:  Att(+2)= 6.417%(998/15552)  Att(+1)=43.583%(6778/15552)  Def(-
1)=31.031%(4826/15552)  Def(-2)=18.969%(2950/15552)

Game value table (rows=attack, cols=defense):

GV =

   -0.6667         0
   -0.4630   -0.2504
   -0.3333   -0.1255

(Att_Hero=0 , Def_Hero=1 , Def_Fort=1)

>>
```

Output 6: Simulation program results for added bonus battle rules (ATT=0 / DEF=+2)

It is interesting to see how this double bonus for the defender makes the attack much more difficult for the attacking force, in relation to all the other configurations. An attack of ratio 1:1 has less than

17% chance of a winning outcome for the attacker, while even a 3:1 attacking configuration is, again, most likely to fail.

<table>
<tr><th></th><th colspan="3">DEF (+2)</th></tr>
<tr><th rowspan="4">ATT (0)</th><th></th><th>(1)</th><th>(2)</th></tr>
<tr><th>(1)</th><td>-0,6667</td><td></td></tr>
<tr><th>(2)</th><td>-0,4630</td><td>-0,2504</td></tr>
<tr><th>(3)</th><td>-0,3333</td><td>-0,1255</td></tr>
</table>

Table 10: Added bonus game rules battle outcome matrix (ATT=0 / DEF=+2)

Using the same game-theoretic analysis for the game matrix produced by the simulation program, it is fairly easy to locate a saddle-point at the cell (3,1), yielding a game value of –0,3333. Here, the game matrix is somewhat different from all the previous situations, due to the high bias towards the defender. Even though the attacker chooses to use the maximum allowable force size, the defender chooses to use only 1 unit instead, securing the added dice bonus of +2 for it and limiting any loss due to unfortunate dice drawing to only a single unit. This is a rather interesting result, since a human player might be tempted to employ more than 1 unit in order to finish off the battle more quickly, however in the long run this choice would probably result in heavier casualties.

As before, it is fairly easy to use the game value to calculate the a-priori winning probabilities for both the attacker and the defender, as well as the equilibrium forces ratio:

$$G_{value} = -0{,}3333 \Rightarrow P_A=50\%+(G_{value}/2)=33{,}333\% \ / \ P_D=50\%-(G_{value}/2)=66{,}667\%$$

$$\mathbf{R_e} = N_A/N_D = P_D/P_A = (66{,}667\% \ / \ 33{,}333\%) = \mathbf{2{,}0}$$

The resulting equilibrium forces ratio shows that, although a single +1 bonus for the defender yielded a +12% force size required by the attacker, here the +2 bonus for the defender essentially doubles the defending force capability in terms of battle strength. If an attacker is to overrun such a defending force, an attacking force of at least double this size has to be used in the battle.

The following matrix demonstrates the differences on the probable battle outcomes, in contrast to the same matrix calculated over the standard battle rules:

A\D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	70,37%	40,74%	20,99%	10,01%	4,53%	1,97%	0,83%	0,34%	0,14%	0,05%	0,02%	0,01%	0,00%	0,00%	0,00%
4	80,25%	53,91%	31,96%	17,33%	8,79%	4,24%	1,97%	0,88%	0,39%	0,16%	0,07%	0,03%	0,01%	0,00%	0,00%
5	86,83%	64,88%	42,94%	25,86%	14,48%	7,66%	3,86%	1,88%	0,88%	0,40%	0,18%	0,08%	0,03%	0,01%	0,01%
6	91,22%	73,66%	53,18%	34,97%	21,31%	12,21%	6,64%	3,47%	1,74%	0,85%	0,40%	0,19%	0,09%	0,04%	0,02%
7	94,15%	80,49%	62,28%	44,07%	28,90%	17,77%	10,35%	5,76%	3,08%	1,59%	0,80%	0,39%	0,19%	0,09%	0,04%
8	96,10%	85,69%	70,09%	52,74%	36,85%	24,13%	14,95%	8,82%	5,00%	2,73%	1,44%	0,74%	0,37%	0,18%	0,09%
9	97,40%	89,60%	76,59%	60,69%	44,80%	31,02%	20,30%	12,65%	7,55%	4,33%	2,41%	1,30%	0,68%	0,35%	0,17%
10	98,27%	92,49%	81,89%	67,76%	52,45%	38,16%	26,26%	17,19%	10,76%	6,48%	3,76%	2,12%	1,16%	0,62%	0,32%
11	98,84%	94,60%	86,13%	73,88%	59,59%	45,31%	32,61%	22,33%	14,62%	9,19%	5,57%	3,27%	1,86%	1,03%	0,56%
12	99,23%	96,15%	89,47%	79,08%	66,09%	52,23%	39,15%	27,93%	19,05%	12,48%	7,87%	4,80%	2,84%	1,64%	0,92%
13	99,49%	97,26%	92,06%	83,41%	71,86%	58,78%	45,69%	33,85%	23,99%	16,31%	10,69%	6,77%	4,15%	2,48%	1,44%
14	99,66%	98,06%	94,06%	86,96%	76,89%	64,81%	52,07%	39,92%	29,30%	20,64%	14,01%	9,18%	5,83%	3,59%	2,16%
15	99,77%	98,63%	95,58%	89,83%	81,21%	70,28%	58,14%	45,99%	34,86%	25,38%	17,80%	12,05%	7,90%	5,03%	3,11%

Table 11: Added bonus game rules winning probabilities matrix (ATT=0 / DEF=+2)

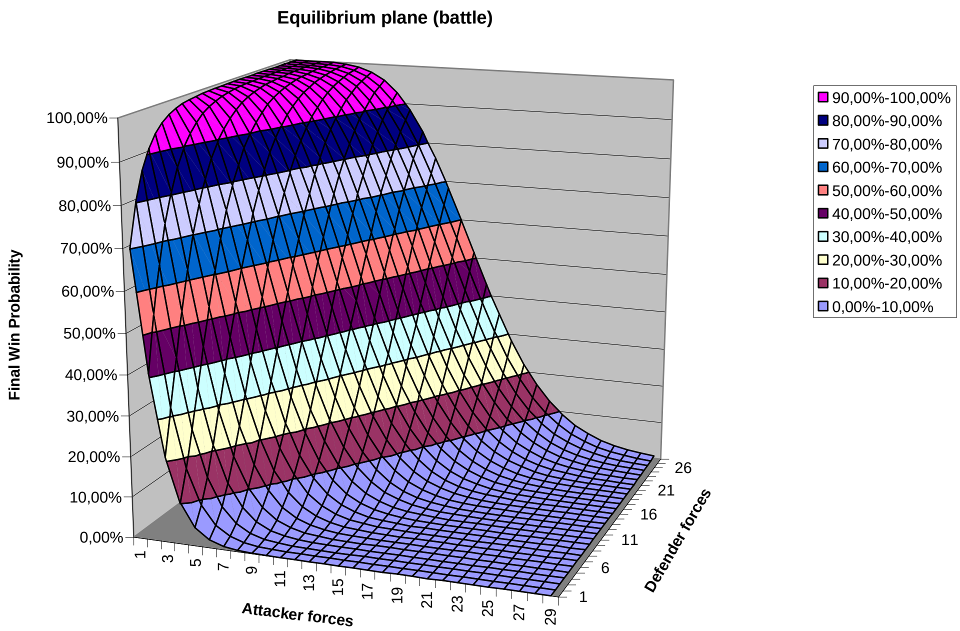

Figure 6: Added bonus game rules winning probabilities 3D plot (ATT=0 / DEF=+2)

8. Playing optimally

The numerous tables and graphs presented above are only a detailed scheme for optimal strategies in various battle rules and situations. However, since it's not very easy, at least for human players, to carry a set of probability tables for every case, nor it is *fun*, as it is supposed to be when playing board games like RISK, a few summarizing notes can be used as guidelines for playing as an expert:

- For standard battle rules (0/0) or when equal "hero" and "fortress" bonuses (+1/+1) apply, the attacker should use at least as much units as the defending force size.
- When a "hero" bonus is available for the attacker and no bonuses are available for the defender (+1/0), the attacker can use 72% or more of the defender's force size.
- When the defender exhibits a +1 difference in bonuses towards the attacker, i.e. when the attacker has no "hero" pawn versus one defender's bonus (0/+1) or has a "hero" pawn versus two defender's bonuses (+1/+2), at least 12% larger force should be used during the attack.
- Finally, when the defender has both a "hero" and a "fort" in the area, the attacker, in case of an attack, should use at least double the size of the defending force.

The following table can quantitatively describe these simple guidelines:

Battle Config.	Attacker's Force	Attacker's Bonus	Defender's Force	Defender's Bonus	Game Value	Attacker Win Prob(%)	Defender Win Prob(%)	Equilibrium Forces Ratio (A/D)
1: (0/ 0)	3	0	2	0	+0,0395	51,975%	48,025%	**0,924**
2: (+1/ 0)	1	+1	1	0	+0,1667	58,335%	41,665%	**0,7142**
3: (+1/+1)	3	+1	2	+1	+0,0395	51,975%	48,025%	**0,924**
4: (+1/+2)	3	+1	2	+2	–0,0566	47,170%	52,830%	**1,12**
5: (0/+1)	3	0	2	+1	–0,0566	47,170%	52,830%	**1,12**
6: (0/+2)	3	0	1	+2	–0,3333	33,333%	66,666%	**2,0**

Table 12: Summary of the obtained results on optimal RISK battle strategies

As already stated previously, for every board game like RISK, where the conquer and control of areas in a "map" are a matter of outmost importance for winning the overall game, solving the game at the tactical (lower) levels effectively transforms the task of strategic planning into a logistics problem for optimal forces allocation. When a player knows how to safely secure an area towards any upcoming attack or launch an attack with the most promising chances of success, the optimal allocation of forces in the overall region can ensure a very strong strategic stance in any game setup. Game-theoretic models of estimating optimal strategies, extended with analytical probabilistic models of outcome estimation, can form a solid base for any higher-level automatic A.I. planning like a robust *graphsearch* algorithm (Minimax, A-B, etc).

9. Conclusion

Game Theory is a useful theoretical mathematic tool when dealing with robust models of nearly every type of situations of conflicting interests, whether it is a simple board game like RISK, scheduling the routes of public transportation networks or optimizing a satellite's trajectory versus fuel consumption. A framework of mathematical theory that started nearly 60 years ago, quite recent it terms of the complete history of Mathematics, has become a controversial tool in times of war, cold war and peace. It is no coincidence that the famous RAND corporation, an unidentified (at that time) yet viable part of the Advanced Research Projects Agency of the Department of Defense of the USA during the Cold War, has extensively used advanced game-theoretic modeling and simulation to optimize the defensive and offensive nuclear grid of the USA, including some rather extreme and somewhat peculiar studies that pointed the way towards transferring the hypothesized intercontinental conflict into space. Today, RAND Corporation is a commercial company, offering its sophisticated services in logistics, planning and defense studies in various research areas. Fortunately, the only studies that have been extensively tested over the last few years are mostly related with Economics and Logistics.

10. Final notes

This report is an extensive version of an earlier packet of notes, produced over a brief discussion over the Usenet during May 2002. A related message, including the Matlab source code for the RISK battle outcome simulation, were first posted at comp.ai.games on 14th May 2002 under the alias xgeorgio@ieee.org. Since then I received numerous messages requiring further explanations about the Matlab source code and the produced results. As the program itself is nothing more than a dice-drawing enumerator, I decided it was better to explain the overall simulation methodology instead, presenting a practical example of applied game-theoretic modeling and giving a brief glimpse of the underlying theory behind it. The source code itself is quite generic and can be executed nearly in every version of Matlab, however it has been tested only on versions 4.x and newer.

References

1. Von Neuman, J, and Morgenstern, O, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, 1947.
2. Rapoport, A, Guyer, M J, Gordon, D G, *The 2x2 Game*, The University of Michigan PressAnn Arbor, 1976.
3. Davenport, W C, Jamaican Fishing Village, *Papers in Carribean Anthropology*, 59, Yale University Press, 1960.
4. Davis, M D, *Game Theory*, Basic Books, New York, 1983.
5. Dresher, M, *The Mathematics of Games and Strategy, Theory and Applications*, Dover, New York, 1981.
6. Jones, A J, *Game Theory: Mathematical Models of Conflict*, Halstead Press, Chichester, England, 1980.
7. Maynard Smith, J, and Parker, G A, The Logic of Animal Conflict, *Nature*, 246(1973) , 15-18.
8. Stahl, S, *A Gentle Introduction to Game Theory*, Mathematical world vol.13, American Mathematical Society, 1998.
9. Dresher, M, Games of Strategy, Theory and Applications, RAND Corp (Math.Res.Dept.), Prentice-Hall, Eaglewood Cliffs, NJ, 1961.
10. Usenet (NNTP), Artificial Intelligence and Games newsgroup at comp.ai.games.

```
%------------------------------------------------------------------------------------
%
%   PROJECT:        -
%   PACKAGE:        -
%   FILE:           "risk3.m"
%
%   PURPOSE:        RISK board game - cost analysis & evaluation
%   VERSION:        1.2
%   PLATFORM:       MathWorks MATLAB (versions 4.x and later)
%
%   STAGE:          RELEASE
%   UPDATED:        02-Jan-2004/15:00
%   HISTORY:
%       version 1.2: second release version completed
%       version 1.1: added "Hero" and "Fortress" bonus options
%       version 1.0: corrected cummulative game value calculation
%       version 0.9: first release version (beta) completed
%       version 0.7: added contour plot when finished
%       version 0.5: core completed, results verified
%
%   DESCRIPTION:
%                  This program calculates all possible outcomes in standard
%                  RISK game and evaluates the gain/cost for each player.
%                  Attack player is positive-valued while defense player is
%                  negative-valued. The outcome table can be used to construct
%                  optimal forces confrontations (attack:defense ratio).
%
%                  For standard game rules (i.e. no "Hero" or "Fortress" bonuses),
%       results table shows that attacker has a preferred strategy
%       at N=3 (max) and defender has a        preferred strategy at N=2 (max),
%       both derived as strictly dominative strategies. The saddle-point
%       is at A=3/D=2 and the game value is v=0.0395, i.e. slightly biased
%       towards the attacker.
%
%              |   Def(1)    Def(2)
%       -------+-------------------
%       Att(1) |  -0.1667         0
%       Att(2) |   0.1574   -0.1103
%       Att(3) |   0.3194    0.0395
%
%       In any case, the game value GV(i,j) can be used to derive the proportional
%       support probabilities for each side, using pA=0.5+GV(i,j) for the
%       attacker and pD=0.5-GV(i,j) for the defender. The resulting values
%       can then be used to calculate the appropriate forces ratios for an
%       equilibrium, i.e. a draw in battle. For GV(3,2), which is the
%       saddle-point in this situation, pA=0.5395 and pD=0.4605, so the forces
%       ratio for an equilibrium in battle is:
%           fA*pA = fD*pD  =>  fA/fD = pD/pA = 0.8536
%       Based on this ratio, a full forces opposition matrix can be
%       calculated for battles of configuration A:D=3:2, which is the
%       preferred strategy for both players. Using a ceiling function for
%       the calculation of forces (integer), close-win situations for the
%       attacker are:
%           A:D = 3:2, 3:3, 4:4, ... 9:10, ..., 18:20, ..., 26:30, ...
%       The slightly lower forces required by the attacker to overcome the
%       defender is due to the fact that in the specific (saddle-point)
%       situation of 3:2 battles, the positive game value gives a slight
%       advantage to the attacker's side. Any player exhibiting forces ratio
%       biased towards its side has the initiative of winning the battle.
%       Although each battle round is independent from each other, initial
%       forces ratio at the start of the battle should give a very good hint
%       about the most probable final outcome in the long run.
%
%       The "Hero" and "Fortress" options for battle bonuses apply for the
%       RISK rules presented in "The Lord of the Ring" game variance. It
%       adds some extra points (+1 or +2) in the attacker's and/or
%       defender's maximum dice numbers (only) when moving into battle with the
%       support of a hero character and/or a fortress.
%
%   AUTHOR: Harris Georgiou - xgeorgiou@yahoo.com
%   LICENCE: Mozilla Public Licence version 1.1
%   COPYRIGHT: Harris Georgiou (c) 2004
%
%------------------------------------------------------------------------------------

disp('RISK board game - evaluation of outcomes, version 1.2');
disp('Harris Georgiou (c) 2004, mailto:xgeorgiou@yahoo.com');
```

```
disp(' ');

msg=sprintf('calculating...\n');
disp(msg);

clear all;

SD=6;       % size of dices

heroA=0;    % 0 or +1 (only "Hero" bonus for the attacker)
heroB=0;    % 0 or +1
fortB=0;    % 0 or +1


% ... Attack=1, Defense=1 ...
G11=[];  resA1=0;  resD1=0;
for Ai=1:SD,
    for Di=1:SD,
        if ( Ai+heroA <= Di+heroB+fortB ),    % compare attack/defense draws (equal => defense
wins)
            G11=[G11 -1]; resD1=resD1+1;
        else
            G11=[G11 +1]; resA1=resA1+1;
        end;
    end;
end;
gv11=(resA1-resD1)/size(G11,2);     % use analytical formula for game value (generic)

t=[resA1/size(G11,2)*100 resD1/size(G11,2)*100];    % calculate partial percentages
msg=sprintf('GAME 1x1 PROB:  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)',t(1),resA1,size(G11,2),t(2),resD1,size(G11,2));
disp(msg);

% ... Attack=2, Defense=1 ...
G21=[];  resA1=0;  resD1=0;
for Ai=1:SD,
    for Aj=1:SD,
        A=sort([Ai Aj]);    % sort (ascending) the 2 attack draws
        for Di=1:SD,
            if ( A(2)+heroA <= Di+heroB+fortB ),  % compare attack/defense draws (equal =>
defense wins), use first max value
                G21=[G21 -1]; resD1=resD1+1;
            else
                G21=[G21 +1]; resA1=resA1+1;
            end;
        end;
    end;
end;
gv21=(resA1-resD1)/size(G21,2);     % use analytical formula for game value (generic)

t=[resA1/size(G21,2)*100 resD1/size(G21,2)*100];    % calculate partial percentages
msg=sprintf('GAME 2x1 PROB:  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)',t(1),resA1,size(G21,2),t(2),resD1,size(G21,2));
disp(msg);

% ... Attack=3, Defense=1 ...
G31=[];  resA1=0;  resD1=0;
for Ai=1:SD,
    for Aj=1:SD,
        for Ak=1:SD,
            A=sort([Ai Aj Ak]);    % sort (ascending) the 3 attack draws
            for Di=1:SD,
                if ( A(3)+heroA <= Di+heroB+fortB ),     % compare attack/defense draws (equal
=> defense wins), use first max values
                    G31=[G31 -1]; resD1=resD1+1;
                else
                    G31=[G31 +1]; resA1=resA1+1;
                end;
            end;
        end;
    end;
end;
gv31=(resA1-resD1)/size(G31,2);     % use analytical formula for game value (generic)

t=[resA1/size(G31,2)*100 resD1/size(G31,2)*100];    % calculate partial percentages
msg=sprintf('GAME 3x1 PROB:  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)',t(1),resA1,size(G31,2),t(2),resD1,size(G31,2));
```

```
disp(msg);

% ... Attack=1, Defense=2 ... (=> not a valid combination)
%G12=[];  resA1=0;  resD1=0;
%for Ai=1:SD,
%    for Di=1:SD,
%        for Dj=1:SD,
%            D=sort([Di Dj]);    % sort (ascending) the 2 defense draws
%            if ( Ai+heroA <= D(2)+heroB+fortB ),      % compare attack/defense draws (equal
=> defense wins), use first max values
%                G12=[G12 -1];  resD1=resD1+1;
%            else
%                G12=[G12 +1];  resA1=resA1+1;
%            end;
%        end;
%    end;
%end;
%gv12=(resA1-resD1)/size(G12,2);     % use analytical formula for game value (generic)

% Note: standard rules forbit A:D setup of 1:2, reset to zero to make neutral
G12=[0];  resA1=0;  resD1=0;        % set matrix cell to zero (not used)
gv12=(resA1-resD1)/size(G12,2);     % use analytical formula for game value (generic)

%t=[resA1/size(G12,2)*100 resD1/size(G12,2)*100];    % calculate partial percentages
%msg=sprintf('GAME 1x2 PROB:  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)',t(1),resA1,size(G12,2),t(2),resD1,size(G12,2));
%disp(msg);

% ... Attack=2, Defense=2 ...
G22=[];  resA1=0;  resD1=0;  resA2=0;  resD2=0;
for Ai=1:SD,
    for Aj=1:SD,
        A=sort([Ai Aj]);    % sort (ascending) the 2 attack draws
        for Di=1:SD,
            for Dj=1:SD,
                tA=0; tD=0;
                D=sort([Di Dj]);    % sort (ascending) the 2 defense draws
                if ( A(2)+heroA <= D(2)+heroB+fortB ),    % use FIRST max values, compare
attack/defense draws (equal => defense wins)
                    G22=[G22 -1];  tD=tD+1;
                else
                    G22=[G22 +1];  tA=tA+1;
                end;
                if ( A(1) <= D(1) ),    % use SECOND max values, compare attack/defense draws
(equal => defense wins)
                    G22=[G22 -1];  tD=tD+1;
                else
                    G22=[G22 +1];  tA=tA+1;
                end;
                if (tD>1), resD2=resD2+1; else resD1=resD1+1; end;
                if (tA>1), resA2=resA2+1; else resA1=resA1+1; end;
            end;
        end;
    end;
end;
gv22=(2*resA2+resA1-resD1-2*resD2)/size(G22,2);     % use analytical formula for game value
(generic)

t=[resA2/size(G22,2)*100 resA1/size(G22,2)*100 resD1/size(G22,2)*100 resD2/size(G22,2)*100];
% calculate partial percentages
msg=sprintf('GAME 2x2 PROB:  Att(+2)=%6.3f%%(%d/%d)  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)  Def(-
2)=%6.3f%%(%d/%d)',t(1),resA2,size(G22,2),t(2),resA1,size(G22,2),t(3),resD1,size(G22,2),t(4),r
esD2,size(G22,2));
disp(msg);

% ... Attack=3, Defense=2 ...
G32=[];  resA1=0;  resD1=0;  resA2=0;  resD2=0;
for Ai=1:SD,
    for Aj=1:SD,
        for Ak=1:SD,
            A=sort([Ai Aj Ak]);     % sort (ascending) the 3 attack draws
            for Di=1:SD,
                for Dj=1:SD,
                    tA=0; tD=0;
                    D=sort([Di Dj]);        % sort (ascending) the 2 defense draws
```

```
                    if ( A(3)+heroA <= D(2)+heroB+fortB ),    % compare attack/defense draws
(equal => defense wins), use first max values
                        G32=[G32 -1];  tD=tD+1;
                    else
                        G32=[G32 +1];  tA=tA+1;
                    end;
                    if ( A(2) <= D(1) ),    % compare attack/defense draws (equal => defense
wins), use second max values
                        G32=[G32 -1];  tD=tD+1;
                    else
                        G32=[G32 +1];  tA=tA+1;
                    end;
                    if (tD>1), resD2=resD2+1; else resD1=resD1+1; end;
                    if (tA>1), resA2=resA2+1; else resA1=resA1+1; end;
                end;
            end;
        end;
    end;
end;
gv32=(2*resA2+resA1-resD1-2*resD2)/size(G32,2);     % use analytical formula for game value
(generic)

t=[resA2/size(G32,2)*100 resA1/size(G32,2)*100 resD1/size(G32,2)*100 resD2/size(G32,2)*100];
% calculate partial percentages
msg=sprintf('GAME 3x2 PROB:  Att(+2)=%6.3f%%(%d/%d)  Att(+1)=%6.3f%%(%d/%d)  Def(-
1)=%6.3f%%(%d/%d)  Def(-
2)=%6.3f%%(%d/%d)',t(1),resA2,size(G32,2),t(2),resA1,size(G32,2),t(3),resD1,size(G32,2),t(4),r
esD2,size(G32,2));
disp(msg);

% ... construct final results table and plot ...
GV=zeros(3,2);
GV=[gv11 gv12; gv21 gv22; gv31 gv32];

disp(' ');
disp('Game value table (rows=attack, cols=defense):');
GV

msg=sprintf('(Att_Hero=%d , Def_Hero=%d , Def_Fort=%d)\n',heroA,heroB,fortB);
disp(msg);

figure(1);
contourf(GV,[0 gv32]);      % draw two contour lines, at levels '0' and at game value
title('RISK - battle outcome evaluation');
xlabel('Defense (ratio)');
ylabel('Attack (ratio)');


%-----------------------------------------------------------------------------------------
```

Last update: 2-Jan-2004 by Harris Georgiou
Contact: xgeorgiou@yahoo.com