Legionnaires: A Combinatorial Game

Matthew D. Boardman
Dalhousie University
Matt.Boardman@dal.ca

April 19, 2005

Abstract

In this paper, we analyze an impartial combinatorial game from which a wide variety of game values may be generated, including integer-valued positions, fractions, nimber-valued positions and infinitesimal positions. The game trees are complex, interesting and have a depth varying at least exponentially with the size of the board. We demonstrate several sequences of values using computer analysis and develop a potential strategy for winning the game. The strategy is incorporated into a publically-available Java applet to test its effectiveness against human players.

1 Introduction

The Pawns on a chess board have always envied their Bishops' ability to move diagonally, and have long admired their King's fighting abilities. Through centuries of training, modern military technology and intense study of the strategies available to them by applying Game Theory, the Pawns have gained the ability to move diagonally like their Bishops, and fight enemy soldiers like their King, all in the time it used to take them to move forward a single square! They are now proud to call themselves Legionnaires of the GreAt roMan Empire.

Over the far, misty hills in the early morning dew, the Legionnaires begin to fight their training battles. The Legatus of each Legion can be heard inspiring his troops, crying out the Legionnaire's Call to Arms: *"Currite sicut episcopi! Pugnate sicut reges!"* (Run like Bishops! Fight like Kings!)

1.1 The Rules of the Game

The game of Legionnaires is played by two players, on a normal 8×8 chess board with twelve pawns available to each player (in practice, the board could be populated with the pawns from two chess sets, or with checkers). The pawns are arranged as shown in Fig. 1, such that for each player, the first three rows are populated by pawns on squares of their own colour. White starts play with the first move, in the tradition of chess.

A move consists of two parts: a player moves a single pawn diagonally by one square or more, as a Bishop moves in chess, and then takes a pawn belonging to the opposing player that is exactly one square away either diagonally or orthogonally, as a King takes in chess. (Move like a Bishop, Fight like a King.) On each move, then, a pawn must be moved at least one space, and an opposing piece must be removed.

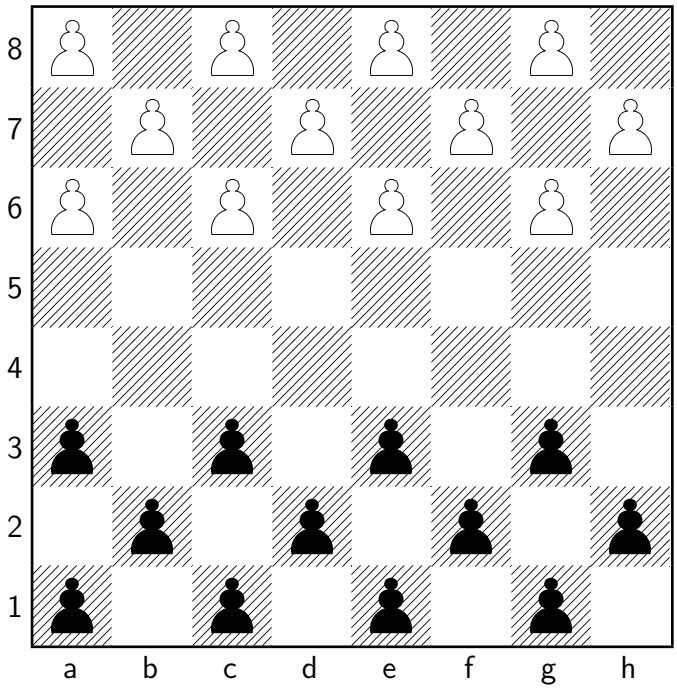


Figure 1: Starting positions for the game of Legionnaires.

Under normal rules, play continues until a player cannot move: if there are no remaining moves available to the player, the player loses the game (a misère convention, in which a player who cannot moves *wins* the game, might be an interesting variation).

2 Small Positions

Let us elaborate on this description by analyzing some small positions, the game values[4] for which can be easily evaluated by hand.

The simplest non-trivial case is a 2×2 board where each player has a single pawn, as shown in Fig. 2. In this case, each player has a single move available: if Black moves first, he will move to the lower right-most position (a single square diagonally), then take the White pawn in the lower left-most position. If White plays first, she has a single, opposite move available with equivalent value. Notice that by convention, we designate White's moves with arrows pointing to the Right, and Black's moves with arrows pointing to the Left. From either of these positions, neither player can move, and their value is therefore 0. This makes the game value of the 2×2 game $\{0|0\}$, where the value of the Black and White options are shown on the left and right respectively, and which by Conway's convention[4] we denote as * ("star").

A slightly more complex position is a 3×3 board where each player has two pawns, as shown in Fig. 3. In this game, each player has four options from the starting position (A):

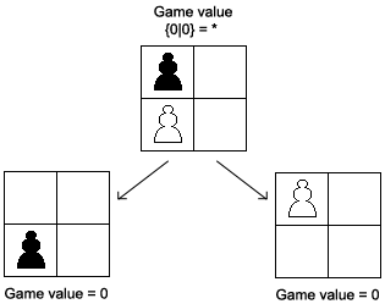


Figure 2: A 2×2 impartial small position that has a game value of *.

either of their pawns can move to the centre of the board and then attack either enemy pawn. In the figure, for simplicity, only two of these eight options are shown: one for each player, (B) and (C). Note that in the analysis of combinatorial games of this type, from every position, we make no assumption as to who's move is next but rather consider both possibilities.

We then expand one of Black's options as an example. From this position (B), Black has two moves available: either of Black's pawns can take the remaining White pawn. From these two resulting positions (D) or (E), neither player can move, so the value of both positions is zero. White also has two options from this position: the one remaining White pawn can remove either of Black's two pawns, resulting in positions (F) or (G). Expanding (F), both players have a single move remaining, to take the enemies opposing pawn, resulting in positions (H) or (I). From these two positions, neither player can move so their game values are zero.

Since (F) allows each player to move to a position worth zero, the value of (F) is $\{0|0\} = *$. We see that since (G) is a mirror image of (F), it must allow the same options to each player, and so its value is also *. Since (D) and (E) both have a value of zero, (B) must have a value of $\{0, 0|*, *\} = \{0|*\}$ (redundant options are removed as they do not affect the value of the game). This particular value is common in infinitesimal values for combinatorial games, so it is defined to be $\uparrow$ ("up") by convention. Similarly, since (C) allows the same options as (B) but favours the White player, its value is $-\uparrow = \downarrow$ ("down").

Finally, from the starting position (A), we see that all eight options are symmetric to either positions (B) or (C). The value of (A) is therefore:

$$\{\uparrow, \uparrow, \uparrow, \uparrow \mid \downarrow, \downarrow, \downarrow, \downarrow\} = \{\uparrow \mid \downarrow\} = *$$

Such simpler cases were used to verify the accuracy of computer calculations, described in the following section.

3 Computer Analysis

The values in Legionnaires can quickly get quite complicated. For example, an impartial 6×6 variation with all six rows populated by three pawns per row has a game value with 699 975 digits and contains well over two million unique positions; to put this number in perspective, if we were to print this position with 80 characters per line and 60 lines per page,

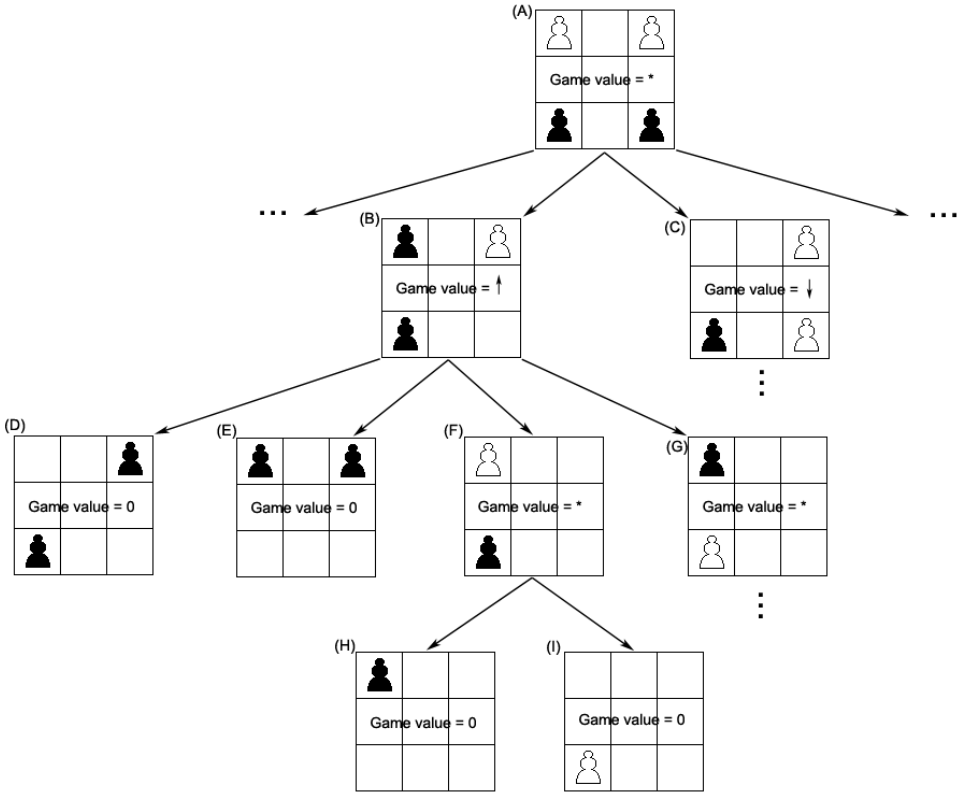


Figure 3: A 3×3 impartial small position that also has a game value of *. From this starting position, 71 positions can be reached—a total of 35 distinct positions.

the game value alone would require 146 pages! Computer analysis is therefore necessary. Even with computer analysis, the game values of many of the positions in this paper take hours or days to complete with current computer hardware.

A Java plug-in for *CG Suite*[6] was written to implement Legionnaires in this environment. The *Pawns1Position* class in this plug-in extends the *AbstractGridGame* abstract class, which handles all the mechanics of interaction with *CG Suite* such that our only real task is to define the options available to a player at any point in the game.

This is achieved by Algorithm 3.1. From a given position, for each pawn of the current player's colour, the computer will try to move as far as possible in each diagonal direction until either reaching the edge of the board or being blocked by another piece. From each position along these paths, the computer checks to see if any enemy pieces are in range, and if so adds a corresponding option to the Collection of options for the particular position.

This plug-in was used to enable CG Suite to analyze games of Legionnaires, including the more complex values found in the Museum section. The test machines were a dual 1.26 GHz Pentium III *Tualatin* with 2GB of 266 MHz DDR memory running Mandrake Linux 10.0 with kernel 2.6.3-07, and a 3.4 GHz Pentium IV-HT laptop with 1GB of dual-channel 400 MHz DDR memory running Microsoft Windows XP Service Pack 1. CG Suite was allowed 1850 MB on the Linux machine and 1492 MB on the Windows machine, through

Algorithm 3.1: GETOPTIONS($Player$)

$OptionsList \leftarrow \emptyset$
for each $Position \in$ this.Grid
do $\begin{cases} \textbf{if } Position = Player \\ \quad \textbf{then } \begin{cases} \textsc{Move}(Position, Northwest) \\ \textsc{Move}(Position, Northeast) \\ \textsc{Move}(Position, Southwest) \\ \textsc{Move}(Position, Southeast) \end{cases} \end{cases}$
return ($OptionsList$)

procedure MOVE($Position, Direction, Player$)
$Midpoint \leftarrow Position + Direction$
while $this.Grid(Midpoint = \emptyset$ **and** $Midpoint \in this.Grid$
do $\begin{cases} \textsc{Take}(Position, Midpoint, Player, North) \\ \textsc{Take}(Position, Midpoint, Player, South) \\ \textsc{Take}(Position, Midpoint, Player, West) \\ \textsc{Take}(Position, Midpoint, Player, East) \\ \textsc{Take}(Position, Midpoint, Player, Northwest) \\ \textsc{Take}(Position, Midpoint, Player, Northeast) \\ \textsc{Take}(Position, Midpoint, Player, Southwest) \\ \textsc{Take}(Position, Midpoint, Player, Southeast) \\ Midpoint \leftarrow Midpoint + Direction \end{cases}$

procedure TAKE($Position, Midpoint, Player, Direction$)
$Destination \leftarrow Midpoint + Direction$
if $this.Grid(Destination) \neq Player$ **and** $this.Grid(Destination) \neq \emptyset$
then $\begin{cases} Option \leftarrow this.Grid \\ Option(Position) \leftarrow \emptyset \\ Option(Destination) \leftarrow Player \\ Add\ Option\ to\ OptionsList \end{cases}$

the use of the **-Xmx** option. Both machines used the Sun Java Runtime Environment (JRE) version 5.0 update 2, the latest available at the time of this writing.

3.1 Stand-Alone C Program

A stand-alone version of the plug-in, written in C language, was also developed using the same algorithm to determine options. The purpose of this was to allow a minimum of system memory to be used, in order to evaluate larger positions, by storing all positions on disk in a large text file. When employed to analyze the 6×6 version, however, this program ran for over two weeks and generated only 300 000 positions (in a 180 MB text file). This performance would be insufficient to measure a larger position in a reasonable amount of time, although it would theoretically be able to complete any position, regardless of size, if left long enough: the positions would be generated at a slower and slower rate, as the size of the text file increases thereby lengthening the search for the uniqueness of each found option.

4 Experiments

In this section, we analyze some starting positions and sequences, and display interesting phenomena found during these analyses.

4.1 Starting Positions

Starting positions for 2×2, 2×3, 3×3, 4×4 and 5×5 have been fully analyzed. The possible positions that can be reached from each of these starting positions have been generated using the computer analysis described above, and the game values including all subpositions are available for download at the author's web site [http://www.cs.dal.ca/~boardman/leg.html].

Analyses for 6×6 (with 18 total pieces) and 8×8 (with 24 total pieces) starting positions are currently in progress. As of this writing, over 4.67 million unique positions have been generated for the former, and 33.55 million unique positions were generated from the latter before the 2GB machine ran out of Java heap space.

The following table summarizes the analyses made thus far.[1] In all cases, the positions are impartial, with an equal number of pawns available to each player, arranged symmetrically (vertically mirrored). All known game values in this table are infinitesimal. These starting positions appear in Fig. 4.

Starting Position	Fig.	Pieces	Num. Positions	Game Value
2×2	4(a)	2	3	*
2×3	4(b)	4	35	*
3×3	4(c)	4	35	*
4×4	4(d)	8	982	(90 chars)
5×5	4(e)	10	-	0
5×5	4(f)	12	14 789	(1632 chars)
6×6	4(g)	12	-	0
6×6	4(h)	18	4.67M+	(699 975 chars)
8×8	1	24	33.55M+	-

[1] Note that some numbers are not currently available due to time and memory constraints.

These starting positions are examined again in Section 4.4 where we examine the question of NP-Hardness.

4.2 Museum of Interesting Positions

We have already seen game values of 0, *, ↑ and ↓ for Legionnaires positions. In Figs. 5–7, we see a variety of weird and wonderful position values, including integer-valued positions, fractions, nimber-valued positions and infinitesimal positions, which appeared during our analysis of more complex positions.

In Fig. 5, we examine some infinitesimal values: values that are infinitesimally close to zero, but not equal to zero. Many such values are possible in Legionnaires: it appears that that most impartial positions are infinitesimal.

In Fig. 6, we find some positions called "switches", which may favour either the Black or the White player depending on who moves first. In (a) we see an example of an infinitesimal switch, and in (c) we see a switch with a "temperature" (the difference between the two values divided by 2) of 1. The example in (b) would not be classified as a switch, as if favours the Black player with a temperature of zero, but its value is important as either player can move to a position worth the same value, favouring Black by 1. We also see some positions with integer game values 1, 2 and 4: this last value was obtained playing an impartial 8×8 variation in which all eight rows are populated by four pawns.

In Fig. 7, we see some positions with fraction values. An excellent discussion of fraction-valued games appears in [3] Ch.1, in which Red-Blue Hackenbush positions are examined. The final diagram (f) in this figure shows a position with a nimber value $\{0, *|0, *\} = *2$: such values will be examined more closely in section 4.6.

4.3 Sequences

Some sequences of positions are also interesting. Small sequences in combinatorial games are often useful in determining the behaviour of a game, and for clues to determining proofs of that behaviour. For example, we have seen how quickly Legionnaires positions can grow from simple values such as 0 or * to complex game values. The $3 \times n$ sequence in Fig. 8 shows an excellent example of this, as shown in the following table:

n	Game Value
1	0
2	*
3	*
4	*
5	$\pm(*, \uparrow)$
6	$\pm(*, \uparrow)$
7	(59 137 171 chars)

To put this number for $n = 7$ in perspective: the Complete Works of William Shakespeare has approximately 1230 pages.[1] If the above game value was printed and bound, at 80 characters per line and 60 lines per page, it would amount to more than ten of these volumes.

This was the most complex value found during our analysis in this paper: but why the sudden jump in complexity in comparison to $n = 6$? It is interesting to note that the value

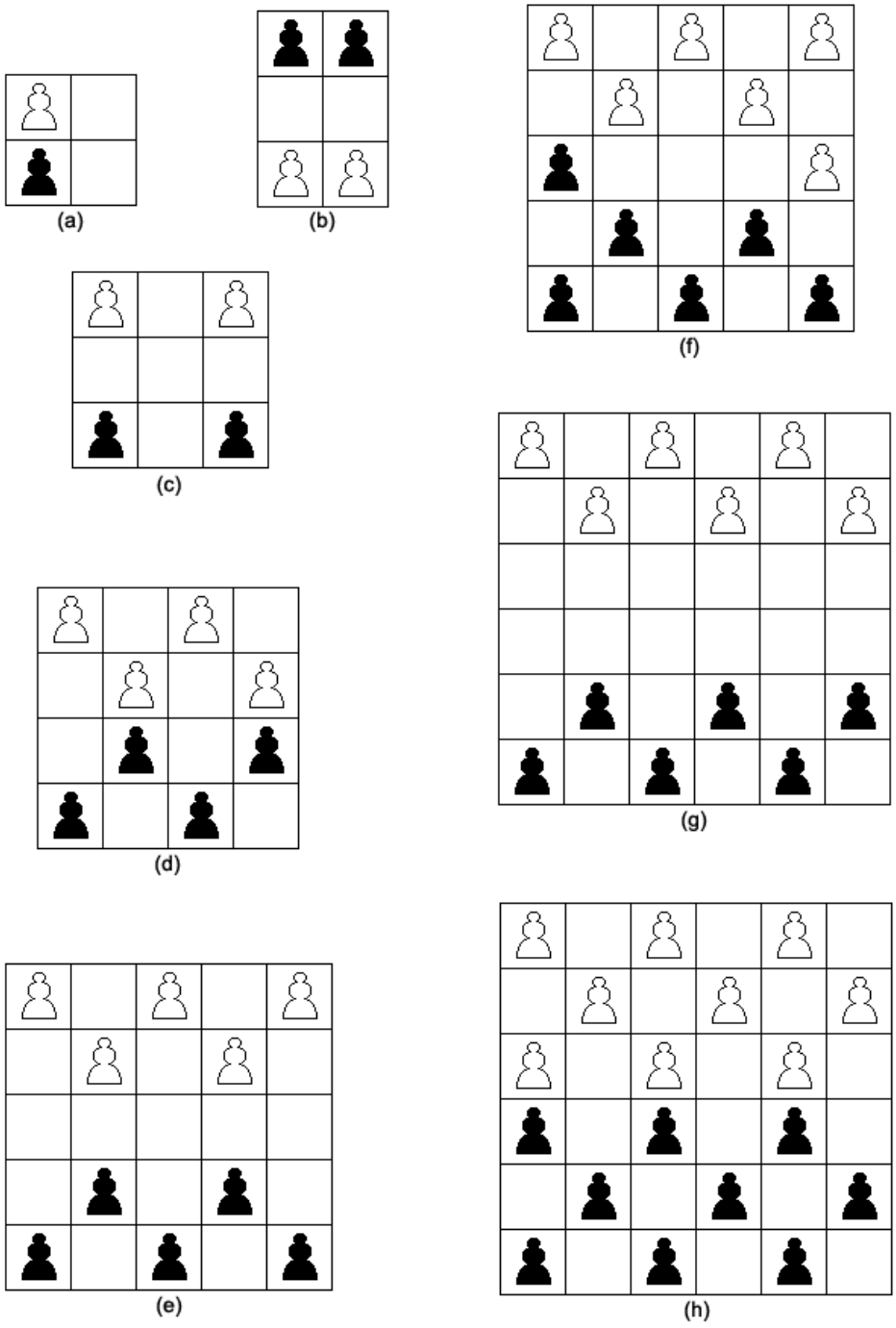
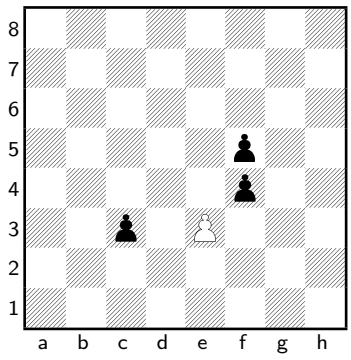
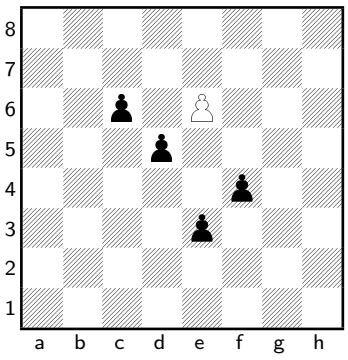


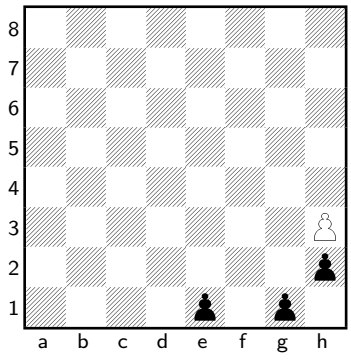
Figure 4: Alternative starting positions.



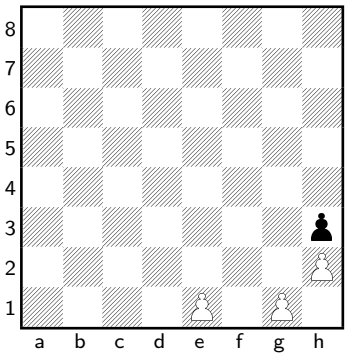
(a) $\Uparrow *$ ("double-up star")



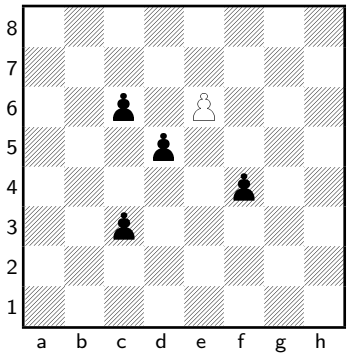
(b) $\uparrow 3$ ("triple-up")



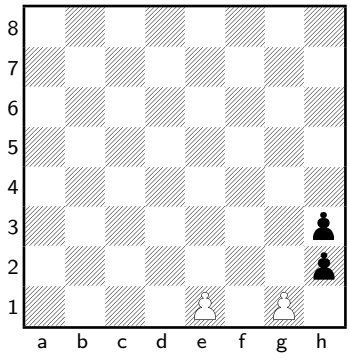
(c) $+_1$ ("tiny 1")



(d) $-_1$ ("miny 1")

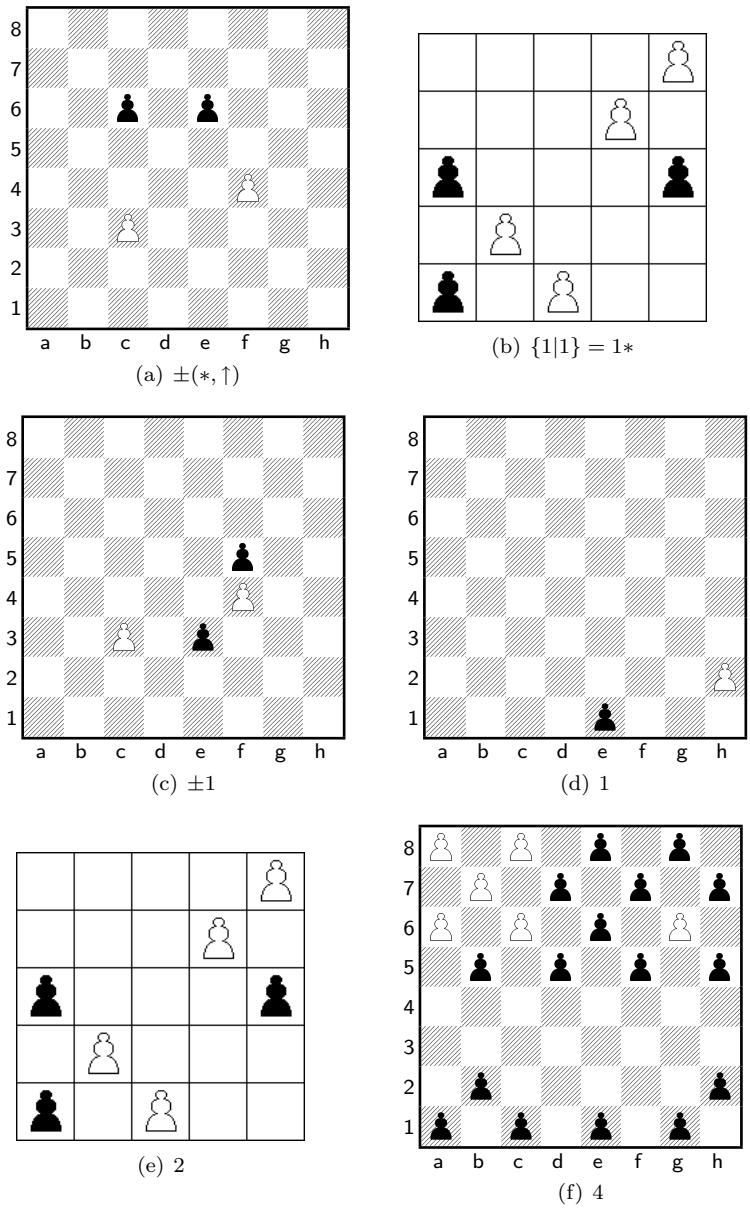


(e) $\{0 \mid \downarrow *\} = \uparrow^2$



(f) $\{* \mid \downarrow\} = \downarrow_{-2}$

Figure 5: Museum of values: infinitesimals.



(a) $\pm(*, \uparrow)$

(b) $\{1|1\} = 1*$

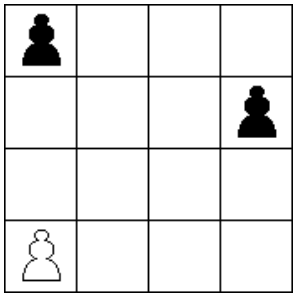
(c) ± 1

(d) 1

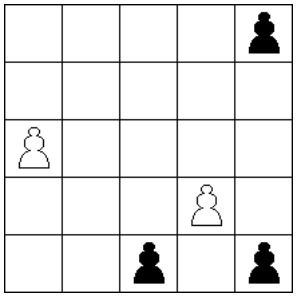
(e) 2

(f) 4

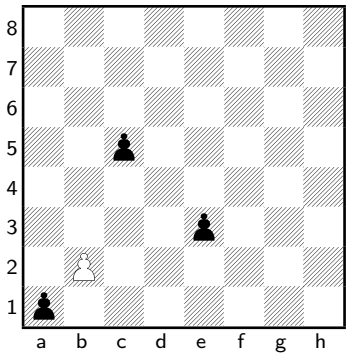
Figure 6: Museum of values: integers and switches.



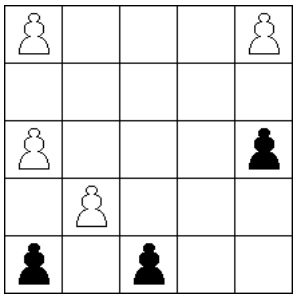
(a) $\frac{1}{2}$



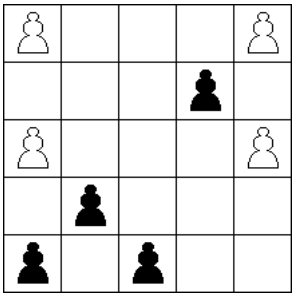
(b) $\frac{3}{4}$



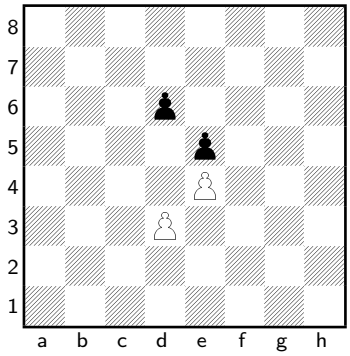
(c) $\frac{1}{4}$



(d) $-\frac{3}{8}$



(e) $-\frac{3}{2}$



(f) *2 ("star two")

Figure 7: Museum of values: fractions and a nimber.

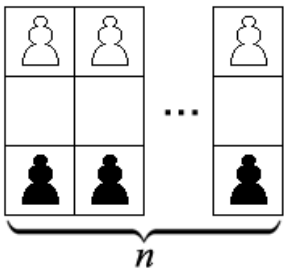


Figure 8: An impartial $3 \times n$ sequence with game values that grow quickly in complexity.

is still infinitesimal and impartial (it begins with a $\pm$) but it's difficult to say more than that about the value. It begins as:

$\pm(\{*|0\},\{\{0|\{\{0|\{0|\{\Uparrow *,\uparrow 3|0,\{\Uparrow *,\{\Uparrow *|0,\pm(*,\uparrow)\}||\pm(*,\uparrow)|0\}\},\{\Uparrow *|\uparrow,\pm(*,\uparrow),\{\Uparrow *,\{\uparrow|*,\pm(*,\uparrow)\}||0|\Downarrow *\}\},\{\Uparrow *,\uparrow 3|\pm(*,\uparrow),\{0<2>|\Downarrow *\},\{\Uparrow *,\{\uparrow|*,\pm(*,\uparrow)\},\{\Uparrow *|0,\pm(*,\uparrow)\}|\downarrow,\{0|\Downarrow *\}\}\},\{\Uparrow *,(\uparrow 3)^2|0,\{\Uparrow *,\{\Uparrow *|0,\pm(*,\uparrow)\}||\pm(*,\uparrow)|0\},\{\{\uparrow|*,\pm(*,\uparrow)\},\{\Uparrow *|0,\pm(*,\uparrow)\}|\downarrow,\{0|\Downarrow *\}\}\},\{(\Uparrow *)^2|\uparrow,\pm(*,\uparrow),\{\Uparrow *,\{\uparrow|*,\pm(*,\uparrow)\},\{\Uparrow *|0,\pm(*,\uparrow)\}|\downarrow,\{0|\Downarrow *\}\}\}||\pm(*,\uparrow),\{\uparrow 3,(\Uparrow *)^2|*,\uparrow,\{\uparrow,\Uparrow *|\downarrow,\{\uparrow,\pm(*,\uparrow)|\Downarrow *\}\},\{\uparrow|*,\{0,*|\Downarrow *\},\{*,\uparrow,\pm(*,\uparrow)|\downarrow,\Downarrow *\}\}||0,\pm(*,\uparrow)|\Downarrow *\}\},\{\{\Uparrow *|\uparrow,\{\Uparrow *,(\uparrow 3)^2|0,\pm(*,\uparrow)\}\},\{0|\{\Uparrow *,\uparrow 3|0,\{\Uparrow *,\{\Uparrow *|0,\pm(*,\uparrow)\}||\pm(*,\uparrow)|0\}\},\{\Uparrow *|\uparrow,\pm(*,\uparrow),\{\Uparrow *,\{\uparrow|*,\pm(*,\uparrow)\}||0|\Downarrow *\}\},\{\Uparrow *,\uparrow 3|\pm(*,\uparrow),\{0<2>|\Downarrow *\},\{\Uparrow *,\{\uparrow|*,\pm(*,\uparrow)\},\{\Uparrow *|0,\pm(*,\uparrow)\}|\downarrow,\{0|\Downarrow *\}\}\},\{\Uparrow *,(\uparrow 3)^2|0,\{\Uparrow *,\{\Uparrow *|0,\pm(*,\uparrow)\}||\pm(*,\uparrow)|0\},\{\{\uparrow|*,\pm(*,\uparrow)\},\{\Uparrow *|0,\pm(*,\uparrow)\}|\downarrow,\{0|\Downarrow *\}\}\},\{(\Uparrow *)^2|\uparrow,\pm(*,\uparrow),\{\Uparrow *,\{$ ... *imagine this continuing for 12 477 pages!*

It would be interesting to parse this value through a lexical parser designed to interpret game values, should such an entity exist, in order to find repeating patterns and perhaps compress these patterns into a new character (such as we compress $\{0||0|-1\}$ into $+_1$ [3, 4]). The resulting simplification may be important for understanding large positions such as this.

4.4 NP-Hardness

A problem is said to be NP-Hard if the number of computations necessary to complete the problem are exponentially-proportional to the length of the input required to define the problem, infinitely often[3, 5]. For example, the evaluation of positions in Red-Blue Hackenbush[3] and the game of Clobber[2] have been shown to be NP-Hard.

Formal proof of NP-Hardness in Combinatorial games is generally not a trivial undertaking [2, 5] and is beyond the scope of this paper. However, we have some indications that Legionnaires may well be NP-Hard.

For example, if we examine the normal starting positions from Figs. 1 and 4, the number of unique subpositions required to fully analyze these positions grows quickly. In fact, if we graph these positions as a function as the number of pieces (the length of the input) as shown in Fig. 9, using a logarithmic scale, we see that the number of subpositions which need to be calculated is approximately proportional to some exponent of the input. The light gray line is a reference $p^{2.3}$, which appears to approximate the curve. Note that the two final points, for 6×6 and 8×8, are only the number of positions that were gathered

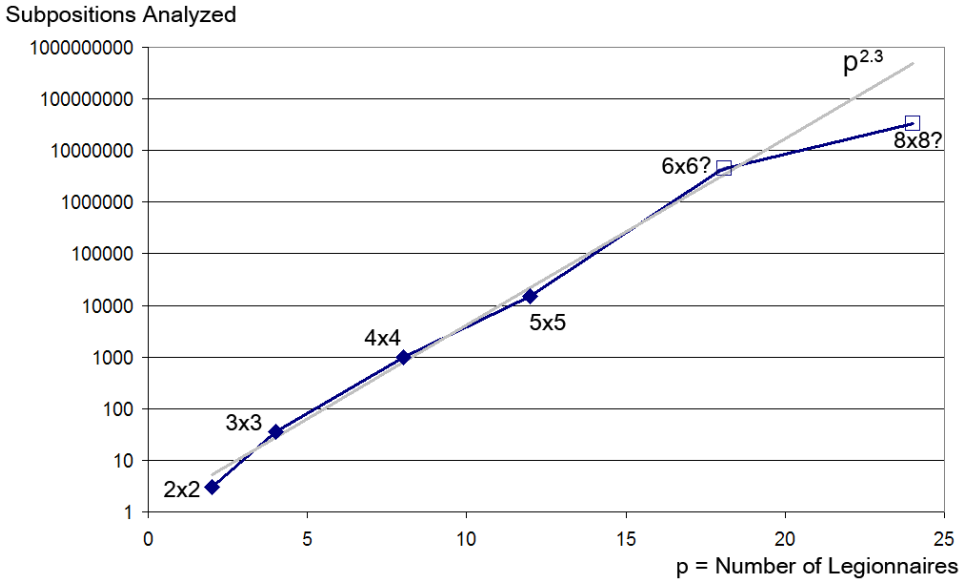


Figure 9: A plot of the number of subpositions required to calculate the game value of impartial starting positions as a function of the number of pieces in the starting position, on a logarithmic scale.

thus far: the actual number of unique subpositions is likely to be much larger, especially for 8×8.

From this it can be seen that as the length of the input n grows, in this case defined by the number of pieces on the board, the number of unique subpositions evaluated in order to completely analyze that position grows approximately exponentially. We therefore have an indication that the problem of evaluating Legionnaires positions is NP-Hard, however this of course is not formal proof.

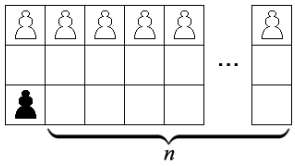


Figure 10: A partizan sequence of n White pawns opposing a single Black pawn.

As a simpler example, if we examine the partizan sequence of a row of n White pawns, an empty row and a single Black pawn as shown in Fig. 10, we see the following values:

n	Game Value
1	0
2	$\downarrow$
3	$\Downarrow *$
4	$\{*, \downarrow \mid 0 \| 0\}$
5	$\{* \mid 0^2\}$
6	$\{*, \downarrow_2 \mid 0 \| 0\}$
7	$\{*, \{*, \downarrow_2 \mid 0\} \mid 0 \| 0\}$
8	$\{\{*, \{*, \downarrow_2 \mid 0\} \mid 0 \| 0\}, \{*, \{*, \{*, \downarrow \mid 0\} \mid 0\} \mid 0\} \mid 0\}$
9	$\{*, \{*, \downarrow_2 \mid 0\} \mid 0 \| 0\}$
10	$\{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0 \| 0\}$
11	$\{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}$
12	$\{*, \{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}$
13	$\{\{*, \{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}, \{*, \{*, \{*, \{*, \{*, \{*, \downarrow \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\}$
14	$\{*, \{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}$
15	$\{*, \{*, \{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}$
16	$\{*, \{*, \{*, \{*, \{*, \{*, \{*, \downarrow_2 \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0\} \mid 0 \| 0\}$

A partially-periodic pattern appears to be developing, and the game value appears to be (roughly) linearly proportional to n. However, if we examine how many positions must be evaluated in order to achieve these values:

n	Positions	n	Positions
2	8	10	1275
3	30	11	2071
4	59	12	3359
5	103	13	5443
6	175	14	8815
7	291	15	14 271
8	479	16	23 099
9	783	17	37 383

The data in this table is plotted in Fig. 11, again on a logarithmic scale. The gray reference curve is 9.5×1.63^p, which appears to closely match the curve when $p \geq 4$. Again, this is an indication of NP-Hardness, but would not be considered to be proof.

4.5 Atomic Weights

Although many game values in Legionnaires are not infinitesimal, many impartial positions are, and this allows us to measure the atomic weight of certain sequences. For example, a $2 \times n$ sequence with a single White Pawn and n Black pawns, all separated by n spaces, where the bottom row is empty is shown in Fig. 12. This sequence has the following values and atomic weights:

n	Value	A.W.	n	Value	A.W.
1	$*$	0			
2	$\uparrow$	1	11	$\uparrow 10*$	10
3	$\Uparrow *$	2	12	$\uparrow 11$	11
4	$\uparrow 3$	3	13	$\uparrow 12*$	12
5	$\uparrow 4*$	4	14	$\uparrow 13$	13
6	$\uparrow 5$	5	15	$\uparrow 14*$	14
7	$\uparrow 6*$	6	16	$\uparrow 15$	15
8	$\uparrow 7$	7	17	$\uparrow 16*$	16
9	$\uparrow 8*$	8	18	$\uparrow 17$	17
10	$\uparrow 9$	9	19	$\uparrow 18*$	18

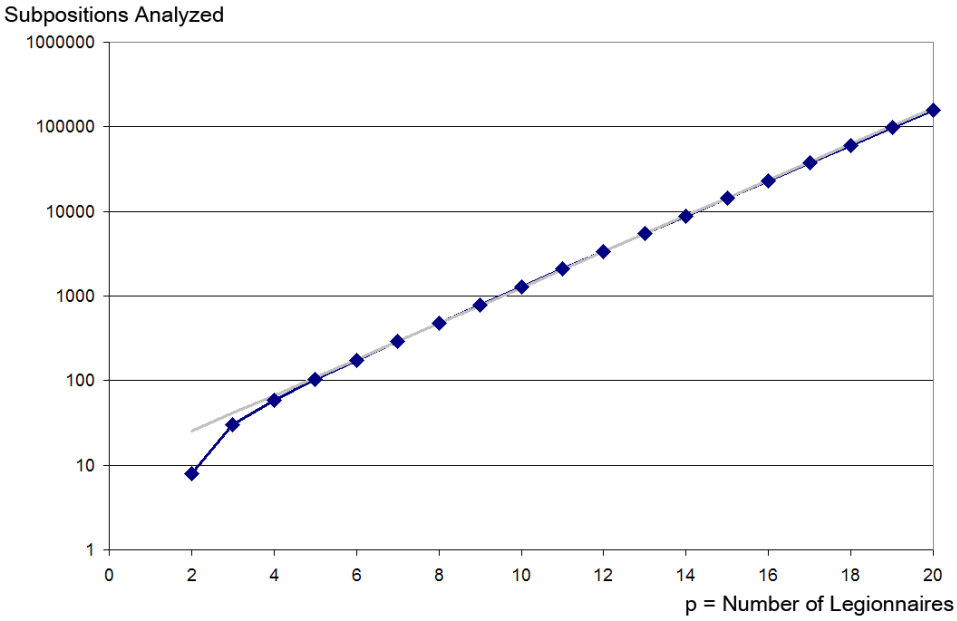


Figure 11: A plot of the number of subpositions required to calculate the game value of the partizan starting position shown in Fig. 10 as a function of the number of pieces in the starting position, on a logarithmic scale.

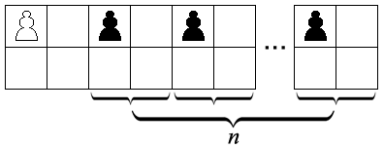


Figure 12: A sequence used to measure atomic weights.

As we can see from this table, many atomic weight values are possible in Legionnaires position. We can conject that this particular sequence appears to have a game value of $\uparrow (n-1) + (* \times n)$ and atomic weight $(n-1)$ for $n \geq 1$.

4.6 Nimber Values

We have seen nimber-valued positions such as 0, $\{0|0\} = *$ and $\{0, *|0, *\} = *2$ in the Museum.

Although $\{0, *, *2|0, *, *2\} = *3$ is currently elusive, with our completed analyses limited to 5×5 boards, we have some clues that it may exist with larger board sizes: for example, the position in Fig. 13 has a value of $\uparrow *3$.

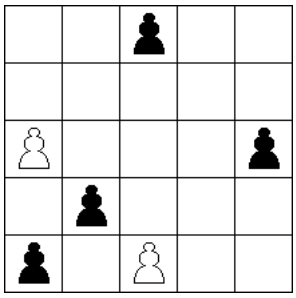


Figure 13: A position of value $\uparrow *3$; not quite a nimber, but a clue that some may appear with further analysis.

5 Strategy

The best strategy found so far appears to be to favour options which will block enemy pawns that are behind the removed pawn. Ideally, this blocking should occur near an edge of the board, so that a single pawn can prevent an enemy from moving at all, rather than simply block a single path. This situation is shown in Fig. 14(a).

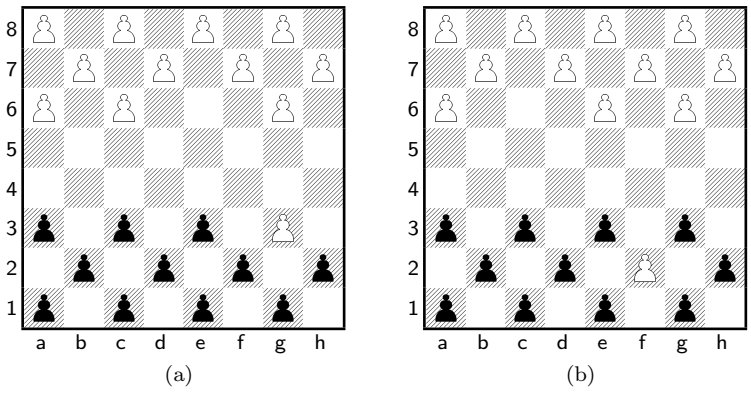


Figure 14: Blocking strategies: (a) An opening move in which White has blocked two enemy pawns. (b) Blocking two pawns in the back row.

In this position, White has made a single opening move from **e6** to **g3**, thereby blocking an avenue of escape for each of enemies **f2** and **h2**. Currently, neither enemy pawn has any available options: **f2**'s other avenue of escape is blocked by **e3**, and **e6** is walled in by the edge of the board.

Making these blocking moves at the beginning of the game has great effect of the remainder of play. It appears logical that blocking the enemy player's middle row of pawns (ie. rows 2 or 7) gives an advantage over blocking the enemy player's back row of pawns (ie. rows 1 or 8) such as in Fig. 14(b), since back row pawns are also at least partially blocked when middle row pawns are blocked. However, further analysis of 8×8 positions would be required to determine this. It has been proposed to enable this addition to the

blocking strategy as a third strategic option in the applet, and allow the computer to play against the standard Black strategy repeatedly to quantitatively measure the worth of this assertion.

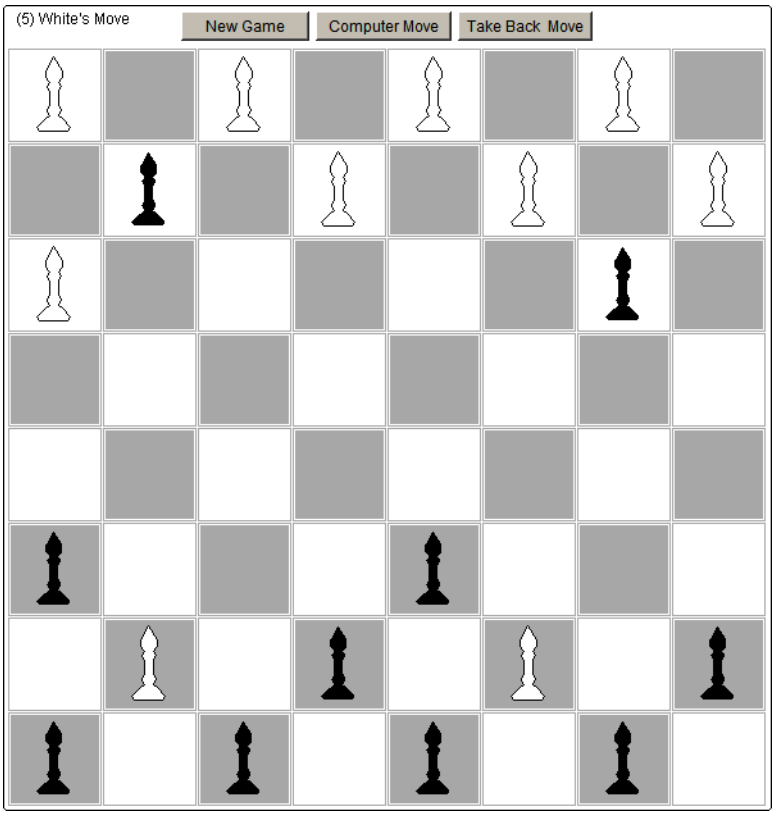


Figure 15: A Java applet developed to allow interested human players to interact with two available strategies for the computer. In this scenario, several blocking moves have been employed by both players.

5.1 Java Applet

A Java applet, shown in Fig. 15, was developed to allow human players to interactively play Legionnaires against the computer. When playing for White, the computer randomly selects one of the available options. However, when playing for Black, the computer will look one move ahead, and elect to play the option from which White will have the fewest number of options available; if there are several of these positions, one is selected at random. This is the thorn in the paw of this strategy: it is possible that one of these "minimal" positions will result in more options being available to White than the other, further along in the game.

The computer often employs the blocking strategy mentioned above, as this results in fewer options being available to White. When a human plays White against the computer's Black strategy, employing this blocking strategy will allow, averaged over many games, a draw. When playing the computer against itself in this applet, Black wins by using this

strategy about eight or nine out of every ten games: sometimes White, playing purely randomly, is lucky!

Some readers may wonder why the Legionnaires in this applet are drawn with vector graphics rather than icons; this is so that the applet can be dynamically resized if running independently rather than embedded on a web page. The proportions of the Legionnaires were chosen to roughly approximate the shape of a Bishop from a Soviet-era Russian chess set, and all dimensions were chosen based on the Golden Ratio $\phi \simeq 1.618$ which appears often in nature.

6 Conclusion

The apparently simple game of Legionnaires is deceptively challenging. Although the rules are simplistic, the game values quickly grow to be quite complex, allowing for integer-values, fractions, nimber-values, infinitesimals and some very complex switches. Several variations may be proposed for further study, including a *Mace* variation in which the player moves diagonally to a midpoint, then attacks all enemy players that are a distance of one square away in any direction (swinging the mace), removing those players from the board (such games are over quickly!). A *Centurion* variation would break the Legionnaires' move into two separate moves, so that a player has the option to *either* move like a Bishop *or* take like a King: this would make the game Loopy, and so its analysis would likely be quite complex; however, the intricate strategies that could arise from this variation may well prove to be interesting.

References

[1] Amazon.com, [http://www.amazon.com], 2005.

[2] M.H. Albert, J.P. Grossman, R.J. Nowakowski, D. Wolfe, *An Introduction to Clobber*, in preparation.

[3] E.R. Berlekamp, J.H. Conway, R.K. Guy, *Winning Ways for Your Mathematical Plays, Vol. 1, 2nd ed.*, A. K. Peters, Ltd., Natick, MA, 2001.

[4] J.H. Conway, *On Numbers and Games, 2nd ed.*, A. K. Peters, Ltd., Natick, MA, 2001.

[5] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[6] A. Siegel, *Combinatorial Game Suite*, [http://cgsuite.sourceforge.net], 2003-2004.

Acknowledgements: This paper was originally written for the graduate mathematics course *Combinatorial and Classical Game Theory* taught by Richard J. Nowakowski at the Department of Mathematics and Statistics, Dalhousie University, who recognized the interestingness and difficulty of this deceptively simple game. Credit is also due to "George from Stoneham, Massachusetts", a regular poster on Able2Know.com's *Latin to English translation* forum, for the Latin translation of the Legionnaire's Call to Arms. Neither bears any responsibility for the inadequacies of this paper.