BLET: A MATHEMATICAL PUZZLE

L. SADUN, F. RODRIGUEZ VILLEGAS AND J. F. VOLOCH

1. The Rules of Blet

Blet is a puzzle. One starts with an even number of coins, laid out in a circle. At first, the coins are laid out with heads and tails alternating (HTHTHT. . . HT). Each turn, you are allowed to take three consecutive coins that show tails-heads-tails and flip them over, getting heads-tails-heads. This increases the total number of heads by one. You may do the opposite, flipping a heads-tails-heads pattern to get tails-heads-tails. The object of the game is to get as many heads as possible. A secondary goal is to reach this maximum in as few moves as possible.

Playing with 4, 6, or 8 coins, it's easy to reach the maximum by being greedy, always converting THT to HTH and never converting HTH to THT. With 10 coins, you can only get 7 heads by being greedy, but there *is* a way to get 8 heads. Can you find it?

In this article we're going to spoil your fun by figuring out what the maximum number of heads is for any starting size, and devising a strategy for reaching that number. Before reading on, you might want to try solving the 10-coin puzzle (Blet-10) on your own. (You may prefer using a 2-color counter, such as are used in *Othello*, instead of coins. Or you can use pencil and paper. An electronic version, with 28 "coins" labeled 0 or 1, is available at http://www.ma.utexas.edu/users/voloch/blet.html).

2. Matrices and polygonal paths

It's inconvenient to work with circular sequences, so we'll pick a starting point, once and for all. Our configuration is then a word w in two symbols H and T, such as the example

$$w = HTHHTTTH \ . \tag{1}$$

If the $k-1$-st, k-th and $k+1$-st letters are THT, we can convert them to HTH. We call this a "type-I" move, and denote it I_k. The reverse procedure, converting HTH to THT, is denoted II_k. In the electronic version, both are done by clicking on the k-th letter,

Date: August 2001.

so we refer to either move as "pushing the k-th button". Note that pushing the first button changes the first, second and last letters, while pushing the last button changes the second-to-last, last, and first letters. We will say that two words w, w' are *equivalent* if we can obtain one from the other by a succesion of type-I and type-II moves.

For example, the word w in (1) and $w' = HHHHTHHT$ are equivalent by the following sequence of moves:

$$\begin{array}{rc} HTHHTTTH & \text{Starting configuration} \\ THTHTTTH & II_2 \\ HHTHTTHT & I_8 \\ HHHTHTHT & I_4 \\ HHHHTHHT & I_5 \end{array} \tag{2}$$

Given a word w we denote by $\ell_H(w)$ its *H-length*, i.e., the number of times that the letter H appears in w; similarly, we let $\ell_T(w)$ be its *T-length*. The total length $\ell(w) = \ell_H(w) + \ell_T(w)$ is of course fixed.

To any word we will associate the movement of a particle in the integer lattice $\mathbb{Z}^2$ as follows. At any given time the particle has two state variables: its *position* q and *momentum* p, which we will think, respectively, as the first and second row of the *state matrix* $M = \binom{q}{p}$. We start at the state $M_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and reading the word w *from right to left* we move from a state $M = \binom{q}{p}$ to the next by the following rules

$$H : \begin{cases} q \mapsto q + p \\ p \mapsto p \end{cases} \tag{3}$$

and

$$T : \begin{cases} q \mapsto q \\ p \mapsto p - q \end{cases} \tag{4}$$

according to the corresponding symbol in w. Note that H does not change the momentum, while T does not change the position. In terms of matrices, let

$$M_H = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \; ; \qquad M_T = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \; . \tag{5}$$

The action of H to to send M to $M_H M$, while T sends M to $M_T M$, so the state matrices always have integer entries and determinant 1.

For any word w, let $\rho(w)$ be the final state matrix. If $w = w_1 w_2$ is a compound word, you should check that $\rho(w) = \rho(w_1)\rho(w_2)$. In technical language, ρ is called a representation into $SL_2(\mathbb{Z})$ of the semigroup of all words in H and T.

For the above example word $w = HTHHTTTH$ we have

$$
\begin{aligned}
M_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
M_1 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = M_H \\
M_2 &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = M_T M_H \\
M_3 &= \begin{pmatrix} -1 & 0 \\ -1 & -1 \end{pmatrix} = M_T M_T M_H \\
M_4 &= \begin{pmatrix} -1 & -1 \\ 0 & -1 \end{pmatrix} = M_T M_T M_T M_H \\
M_5 &= \begin{pmatrix} -1 & -2 \\ 0 & -1 \end{pmatrix} = M_H M_T M_T M_T M_H \\
M_6 &= \begin{pmatrix} -1 & -3 \\ 0 & -1 \end{pmatrix} = M_H M_H M_T M_T M_T M_H \\
M_7 &= \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix} = M_T M_H M_H M_T M_T M_T M_H \\
M_8 &= \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} = M_H M_T M_H M_H M_T M_T M_T M_H = \rho(w) \ .
\end{aligned} \tag{6}
$$

We may represent graphically the movement of the particle by drawing in the plane a segment joining one position vector to the next (a program that draws the pictures given a word is available at http://www.ma.utexas.edu/users/villegas/nmx.html). We obtain in this way a polygonal path in the plane which we will denote by $\mathcal{Q}(w)$. In the same way, we obtain a polygonal path $\mathcal{P}(w)$, in a different plane, by joining one momentum vector to the next. For our running example (1) these look as follows.

FIGURE 1.
The path $\mathcal{Q}(HTHHTTTH)$

FIGURE 2.
The path $\mathcal{P}(HTHHTTTH)$

We will say that a word w is *closed* if the last state matrix coincides with the initial matrix M_0. Geometrically, w is closed if both $\mathcal{Q}$ and $\mathcal{P}$ are closed paths. Algebraically, w is closed if $\rho(w)$ is the identity matrix. We will say that w is *eventually closed* if some repetition $ww\cdots w$ is a closed word. This is equivalent to some power of $\rho(w)$ equaling the identity matrix. You should check that $HTHTHTHTHTHT$ is closed, and that HT and HTH are eventually closed.

Our first goal is to prove the following.

Theorem 1. *Let w be an eventually closed word. Then*

$$\frac{1}{6} < \frac{\ell_H}{\ell} < \frac{5}{6} \qquad \textit{and} \qquad \frac{1}{6} < \frac{\ell_T}{\ell} < \frac{5}{6} \tag{7}$$

In order to prove this theorem we will relate the lengths ℓ_H, ℓ_T and ℓ of a word to geometric data of the path $\mathcal{Q}$.

First we relate ℓ to the winding number of $\mathcal{Q}$; the following formula was proved in [PV] in a somewhat different formulation. Recall that a closed path γ in $\mathbb{R}^2 \setminus \{0\}$ has a well defined *winding number* $m(\gamma)$, which measures how many whole turns it makes around the origin in the counterclockwise direction.

Theorem 2. *Let w be a closed word of total length ℓ and let $\mathcal{Q} = \mathcal{Q}(w)$ be the associated path as defined above. Then*

$$\ell = 12m(\mathcal{Q}) \tag{8}$$

We should remark that both paths $\mathcal{Q}$ and $\mathcal{P}$ have the same winding number.

Now we consider the vertices of $\mathcal{Q}$; by a *vertex* we mean a position vector at which the path $\mathcal{Q}$ changes direction. By construction, a vertex correspond to substrings $T\cdots T$ in w. Let $v_1, \cdots, v_r$ be the vertices of $\mathcal{Q}$ numbered consecutively as we traverse the path. Let θ_j be the corresponding *exterior angle* at the vertex v_j, i.e., the change of angle in $\mathcal{Q}$, measured in the counterclockwise direction, as it comes in and out of v_j. See Figure 3.

FIGURE 3. Exterior angles add up to $2\pi m(\mathcal{Q})$

It is not hard to see that

$$\sum_{j=1}^{r} \theta_j = 2\pi m(\mathcal{P}) \ . \tag{9}$$

Combining this with the above theorem we obtain

$$\sum_{j=1}^{r} \theta_j = \frac{\pi}{6}\ell \ . \tag{10}$$

Since each exterior angle θ_i is strictly less than π, the number r of such angles is strictly greater than $\ell/6$. Since there are one or more T's per vertex, the number of Ts is greater than $\ell/6$. A similar argument concerning the path $\mathcal{P}$ shows that the number of Hs is greater than $\ell/6$, and the theorem is proved.

3. Optimal Blet Configurations

You may be wondering what the last theorem have to do with Blet. The starting configuration for Blet is $(HT)^{n/2}$, where n is the number of coins. This is eventually closed, since this pattern repeated six times is $((HT)^6)^{n/2}$, and $(HT)^6$ is closed. To obtain a bound on the best possible Blet score, we just have to prove that all configurations that are equivalent to the starting configuration are also eventually closed.

Recall that a word w is eventually closed if some power of $\rho(w)$ equals the identity. You should check that $M_T M_H M_T = M_H M_T M_H$. This means that pressing the second, third, ..., or second-to-last button doesn't change $\rho(w)$ at all. Pressing the first button does change $\rho(w)$, but only by conjugation. For any subword w_1, $\rho(Tw_1TH) = A\rho(Hw_1HT)A^{-1}$, where $A = M_T M_H^{-1}$. Pressing the last button has a similar effect. In particular if w and w' are equivalent words, then $\rho(w')^k$ equals the identity if and only if $\rho(w)^k$ does. As a result, all legal Blet configurations are eventually closed, and we have proven:

Theorem 3. *In a Blet game with n coins, it is impossible to get $5n/6$ or more heads. In particular, when playing Blet with $6k$ coins you cannot get more than $5k-1$ heads, with $6k+2$ coins you cannot get more than $5k+1$ heads, and with $6k+4$ coins you cannot get more than $5k+3$ heads.*

4. A Winning Strategy

Let $b(n)$ be the maximum number of heads that can be obtained in Blet with n coins, without ever pushing the first or last button. It's easy to see that $b(2) = 1$, $b(4) = 3$, and $b(6) = 4$.

Theorem 4. $b(n+6) \geq b(n) + 5$.

We will prove this theorem shortly, but first let's consider the consequences. Starting with the values of $b(2)$, $b(4)$, and $b(6)$, we get lower bounds for $b(6n+2)$, $b(6n+4)$ and $b(6n)$. However, these lower bounds are exactly equal to the upper bounds given by Theorem 3. Thus the upper bounds are achievable, without ever touching the first or last button.

Corollary 5. *The best possible score in Blet with n coins is exactly $[(5n-1)/6]$, where $[x]$ denotes the greatest integer less than or equal to x.*

We now prove the theorem. We view the $n+6$ letter starting configuration as an n-letter "body" and an $HTHTHT$ "tail". By assumption, we can convert the body into a word with $b(n)$ H's and $n - b(n)$ T's, while leaving the tail alone. The resulting body then either ends with a T, or ends with a T followed by several Hs.

If the body ends with a T, then the entire word ends with $THTHTHT$. By doing moves I_{n+1} and I_{n+5}, we convert those last 7 letters to $HTHHHTH$. There are now $b(n)+5$ heads, $b(n)$ in the first $n-1$ letters and 5 in the last 7 letters.

If the body ends with an H, then we combine a type-I move with a type-II move to transfer the H to the right of the tail:

$$\begin{aligned} H(HTHTHT) &= HHTHTHT \to HTHTTHT \to HTHTHTH \\ &= (HTHTHT)H \ . \end{aligned} \tag{11}$$

We call this combination a "slide". Notice that the slide doesn't change the total number of H's — it just makes an $(HTHTHT)$ unit swap places with an H. If the body ends with a T followed by k H's, we apply the "slide" k times to convert $TH^k(HTHTHT)$ to $T(HTHTHT)H^k$. We can then do two type-I moves to get $HTHHHTH^{k+1}$.

As an illustration, here is a solution for Blet-10:

(12)

$HTHT\ HTHTHT$	Starting position with 4-letter body and 6-letter tail
$HHTH\ HTHTHT$	I_3 acts on body
$HHTHTHTHTH$	Slide H past tail
$HHHTHHHTHH$	I_4 and I_8.

This solution can then be used to solve Blet-16:

(13)

$HTHTHTHTHT\ HTHTHT$	Start with 10-letter body and 6-letter tail
$HHHTHHHTHHHTHTHT$	Manipulate body as above
$HHHTHHHTHHTHTHTH$	Slide one H past tail
$HHHTHHHTHTHTHTHH$	Slide another H past tail
$HHHTHHHHHTHHHTHHH$	Final two type-I moves.

5. Solving Blet on the computer

Since Blet with a fixed number of coins is a finite game, it can in principle be solved by brute force, with the help of a computer. You systematically list all possible configurations, and then pick the configuration that has the most heads. Alternatively, you can make moves at random. You'll wander through the list of possible configurations and eventually hit every one. When you stop hitting new configurations, or when you run out of patience, stop and pick your best to date.

This random-walk approach works for Blet-4 (5 possible configurations) and Blet-6 (8 configurations), but the system gets more complicated quickly. Blet-28 has over 11 million possible configurations, of which only 196 have the maximum (23) number of heads. It would take many billions of turns to explore the whole list by a random walk, and the odds against finding a maximal configuration are huge.

A better method is called *simulated annealing*. We used a simplified form of this method to guess at the maximal number of heads before we actually had the solution. Before explaining this technique, we explain the much simpler greedy algorithm (that doesn't work in this instance!). Starting from our initial position, every time we reach a position, select a random valid move and move there if (and only if) it will increase the score of the position (that is, the number of Hs). For Blet with 28 coins, the greedy algorithm will never go beyond 21 heads. In the simplified form of simulated annealing we first choose a number ϵ, where $0 < \epsilon < 1$. Again, starting from our initial position, we select a random valid move and move there if it will increase the score, but also move there with probability ϵ if it will decrease the score, and repeat. As described, this procedure may go on forever and may get out of a maximum, so one should keep track of the maximum score encountered and put a limit on the number of iterations. This algorithm is only practical in certain circumstances, but it does work for Blet.

Simulated annealing works when the peaks of the local maxima are not very far from passes, in the sense that a few moves away from a local maximum, there is a position that can be moved greedily to a different local maximum. For instance, in Blet-28, "few moves away" means at most five moves (that is an amusing exercise for the reader). An ϵ between 0.2 and 0.3 seems to work best for Blet-28. Blet does have many more local maxima than global maxima, which explains why greedy algorithms won't work. For Blet-28 there are 115929 local maxima out of 11698223 positions, of which 196 are global maxima. Our implementation of Blet-28 as a computer game has a button that will do the simulated annealing for you.

We should mention that in more sophisticated versions of simulated annealing the ϵ may depend on the score, the so-called energy of the system and on the number of iterations, which is related to the so-called temperature of the system. This process is motivated by the metalurgical procedure of annealing in which a metal is initially heated and then left to slowly cool to achieve a low-energy position. If the results are not acceptable, the procedure is repeated.

We now describe in more detail how we counted the total number of positions and obtained some other data. We assume the reader is familiar with some basic notions of graph theory. Blet, like many similar puzzles, can be modeled by graphs, as follows. Let's say we are playing Blet with n coins and initial position $(HT)^{n/2}$. We can construct a graph whose vertices are all possible positions we can reach from the initial position and the edges link positions that are one move apart. We proceed to show that the number of vertices grows exponentially.

Theorem 6. *Let B_n denote the number of configurations of* Blet-n*.* $2^{n/2} \leq B_n \leq 2^n$*.*

Proof. The upper bound is easy. The set of valid Blet-n configurations is a subset of the set of words of length n in the two letters H and T. There are exactly 2^n such words.

We prove the lower bound by simulating Blet-n within Blet-$(n+6)$, and thereby associating eight Blet-$(n+6)$ configurations to each Blet-n configuration. Start with the usual starting point $(HT)^{(n+6)/2}$, which we view as $(HT)^{n/2}(HT)^3$. We push the first n buttons freely, but whenever we push button $n-1$ we also push $n+2$ and $n+5$, whenever we push button n we also push $n+3$ and $n+6$, and whenever we push button 1 we also push $n+1$ and $n+4$. In this way, the first n letters will always be a valid Blet-n configuration, while the last 6 letters will either be $THTHTH$ or $HTHTHT$, depending on whether the nth letter is an H or a T. You should check that pushing buttons n, $n+3$ and $n+6$ is legal in Blet-$(n+6)$ precisely when pushing button n is legal in Blet-n, and similarly for the other combinations.

After achieving a desired Blet-n position for the first n letters, we still have the freedom to vary the last 6 letters. By pressing combinations of buttons $n+2$, $n+3$, $n+4$, and $n+5$, we can get the final six letters to take any of the eight forms: $HTHTHT$, $THTTHT$, $HHTHHT$, $HTTHTT$, $THTHTH$, $HTHHTH$, $TTHTTH$ or $THHTHH$. Thus we associate eight Blet-$(n+6)$ configurations to every Blet-n configuration, so $B_{n+6} \geq 8B_n$.

Since the lower bound $B_n \geq 2^{n/2}$ holds for $B_4 = 5$, $B_6 = 8$ and $B_8 = 37$, it then follows by induction that it holds for all n.

We don't know the exact number of vertices in general but here is some data for small n:

n	# vertices	# global maxima
4	5	2
6	8	3
8	37	2
10	176	5
12	196	4
14	1471	7
16	6885	16
18	5948	9
20	60460	25
22	280600	55
24	199316	24
26	2533987	91
28	11698223	196
30	7080928	70

It appears from the data that the growth rate is considerably closer to our upper bound than our lower bound.

The above data was obtained by constructing a spanning tree for the graph of Blet. A spanning tree of a graph is a subgraph which contains all vertices of the graph and, moreover, is a tree. We used a standard spanning tree algorithm which goes as follows. We start with the initial position and keep track of all visited positions together with the move that first brought us there. If we are in a certain position, we look for a move that will take us to a new position. If there is, we move there and add the new position to our set of visited positions, otherwise we backtrack from our position using the recorded move that first brought us there, except if the position is the initial one. In the case of the initial position, if we get there and cannot move to a new position, we terminate the algorithm. The reader may check that this is algorithm terminates with a spanning tree. To speed up the computation we also use the fact that the graph is invariant under the group of order n generated by a shift of one composed with replacing heads by tails and vice-versa. We then record only a representative for each orbit. Our implementation (in Tcl) is available by anonymous ftp from ftp.ma.utexas.edu

6. Counting and describing the maxima

We have found the maximum score for Blet-n, and we have exhibited a method for achieving this score. For instance, we have constructed an optimal configuration, namely $H^3TH^4TH^3TH^3$, for Blet-16. Is this the only optimal configuration, or are there others? What do they look like?

Since the original Blet-16 configuration, $(HT)^8$, had rotational symmetry, rotations of $H^3TH^4TH^3TH^3$ by even numbers of steps (e.g., $HTH^4TH^3TH^5$) are achievable and optimal. All such configurations have a T somewhere, followed by 4 Hs, a T, 6 Hs, a T, and 3 Hs. In general, we will denote by $(s_1, \dots, s_k)$ any configuration that is a cyclic permutation of $TH^{s_1}TH^{s_2} \cdots TH^{s_k}$. (4,6,3) is optimal for Blet-16. By reflectional symmetry, (6,4,3) is also optimal and achievable. We will soon see why these are the only optimal configurations for Blet-16. Out of 6885 possible Blet-16 configurations, only 16 are global maxima.

From the proof of Theorems 4 and 6, we obtain a procedure for getting optimal configurations for Blet-$(n+6)$ from optimal configurations for Blet-n. We simulate Blet-n on Blet-$(n+6)$ as in the proof of Theorem 4. Once an optimal configuration $(s_1, \dots, s_k)$ has been obtained for Blet-n, use slide moves to bring the $HTHTHT$ tail adjacent to one of the Ts, say the one between H^{s_1} and H^{s_2}. The single T from Blet-n is thereby replaced with a pattern $THTHTHT$ for Blet-$(n+6)$. Two type-I moves then convert that to $THTTTHT$. This is a new optimal configuration for Blet-$(n+6)$, in which the runs of length s_1 and s_2 have been lengthened by one, and a run of length 3 has been inserted in between. In other words, we have proven

Theorem 7. *If $(s_1, s_2, \dots, s_k)$ is an optimal configuration for Blet-n, then $(s_1+1, 3, s_2+1, s_3, \dots, s_k)$ is an optimal configuration for Blet-$(n+6)$.*

For example, (3,5) is an optimal configuration for Blet-10, so (4,3,6) is an optimal configuration for Blet-16. But (5,3)=(3,5), so (6,3,4) is also an optimal configuration for Blet-16. Similarly, (4,4,3)=(3,4,4)=(4,3,4) is an optimal configuration for Blet-14, so (5,3,5,3), (4,3,5,4) and (5,3,4,4) are optimal configurations for Blet-20.

Theorem 8. *Every optimal Blet configuration may be obtained recursively in this way.*

Proof. We will prove a slightly stronger statement, namely that every eventually-closed word of length n in H and T (with n even) that saturates the upper bounds of Theorem 3 is obtained in this way. As a corollary, this shows that every such word is equivalent to either $(HT)^{n/2}$ or to $(TH)^{n/2}$.

Let w be such a word, and suppose that $n \geq 10$. We will show that 1) the Ts in w are isolated, so w takes the form $(s_1, \ldots, s_k)$ for some positive integers $s_1, \ldots, s_k$, 2) none of the s_i's are equal to one or two, 3) at least one of the s_i's is equal to three, 4) w is obtained by our procedure from an eventually-closed word of length $n-6$ that saturates the upper bounds of Theorem 3. By induction, w is then recursively constructed from an optimal eventually-closed word of length 4, 6, or 8. Since all such words are easily seen to be Blet words (up to cyclic permutation), the proof will be complete.

Step 1. Theorem 1 gave a lower bound on the number of vertices in the polygonal path $\mathcal{Q}$. However, each vertex corresponds to a string of Ts. To minimize the number of Ts (i.e., to maximize the number of Hs), we must place exactly one T in each string. In other words, the Ts must be isolated.

Step 2. If s_i were equal to 1, we would have a string THT somewhere, which we could convert to HTH, thereby increasing the number of Hs. This contradicts the fact that w is optimal. Thus s_i cannot equal one. Now suppose $s_i = 2$ and $n \geq 12$, so that w contains at least three Ts. If $s_i = 2$, there exists the pattern HTHHTH somewhere. By pushing the fifth button of this string we convert it to HTHTHT. We then do slide moves to bring this adjacent to a third T, and finally do two type-I moves to convert $THTHTHT$ to $HTHHHTH$. The net result of all these moves is to increase the number of Hs by one, contradicting the optimality of w. The only remaining case is $s_i = 2$ and $n = 10$, i.e., that the pattern is (2,6). However, no power of $\rho(TH^2TH^6) = \left(\begin{smallmatrix} 1 & 4 \\ 0 & 1 \end{smallmatrix}\right)$ equals the identity, so words of the form (2,6) are not eventually closed.

Step 3 requires a lemma:

Lemma 9. *Suppose $s_1, \ldots, s_k$ are integers, none less than four. Then $\rho(TH^{s_1}TH^{s_2}\cdots TH^{s_k})$ is not equal to the identity.*

Proof of Lemma 9. We explicitly compute

$$\rho(TH^{s_i}) = M_T M_H^{s_i} = \begin{pmatrix} 1 & s_i \\ -1 & 1-s_i \end{pmatrix} = -[(s_i-4)X + Y + I] \,, \tag{14}$$

where

$$X = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}, \qquad Y = \begin{pmatrix} -2 & -4 \\ 1 & 2 \end{pmatrix}, \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} . \tag{15}$$

These matrices satisfy the relations

$$Y^2 = 0, \quad X^2 = X, \quad XYX = X, \quad YXY = Y \ . \tag{16}$$

If all the s_i's are equal to 4, then we have $\rho((TH^4)^k) = (-1)^k(I + kY)$, which is not the identity. If any of the s_i's are greater than 4, and none are less than 4, then the coefficient of X in the expansion $\prod_{i=1}^k[(s_i - 4)X + Y + I]$ is strictly positive, so $(-1)^k \prod_{i=1}^k[(s_i - 4)X + Y + I]$ cannot be the identity. This proves the lemma.

Step 3 of Proof of Theorem. If w has the form $(s_1, \dots , s_k)$, with each $s_i \geq 4$, then the lemma states that a certain cyclic permutation of w cannot be closed. But that implies that w cannot be closed. Similarly, applying the lemma to powers of w shows that w cannot be eventually closed.

Step 4. By steps 1, 2, and 3, each optimal eventually closed word is of the form $(s_1, \dots , s_k)$ with each of the s_i's at least 3, and at least one of the s_i's equal to 3. Without loss of generality, we can assume that w is of the form $(s_1, 3, s_3, \dots , s_k)$. w therefore begins with $TH^{s_1-1}HTHHHTHH^{s_3-1}$, which we convert (by two type II moves) to $TH^{s_1-1}THTHTHTH^{s_3-1}$. Now $\rho(HTHTHT) = -I$, so the word w' of length $n - 6$ obtained by replacing $THTHTHT$ with T is eventually closed, and has five fewer Hs than our original optimal word w. Thus w' is an optimal eventually closed word, and w is obtained from w' by the procedure of Theorem 7.

Conclusion of proof: Since steps 1–4 apply to all $n \geq 10$, any optimal eventually-closed word may be obtained by repeated application of the procedure of Theorem 7 to an optimal eventually-closed word of length less than 10, i.e., of length 4, 6 or 8. Up to cyclic permutation, there is only one word of length 4 with only one T, namely TH^3, or (3). Similarly, there is only one word of length 6 with two isolated Ts and no isolated Hs, up to cyclic permutation, namely TH^2TH^2, or (2,2). There are two words of length 8, namely (2,4) and (3,3), but (2,4) is not eventually closed. Thus the only optimal eventually-closed words of length less than 10 are (3), (2,2) and (3,3). All of these are valid Blet configurations, and all longer optimal eventually-closed words are obtained from these by the procedure of Theorem 7. In particular, all optimal eventually-closed words are valid Blet configurations, up to cyclic permutation. This means they are Blet-equivalent either to the original Blet configuration $(HT)^{n/2}$, or to the only other cyclic permutation of this: $(TH)^{n/2}$.

7. Open Problems

It is easy to see that $(HT)^{n/2}$ and $(TH)^{n/2}$ are equivalent when n is divisible by 6 — just push every third button. However, our computer studies show that these configurations are not equivalent when n equals 2, 4, 8, 10, 14, 16, 20, 22, 26 or 28. This leaves two possiblities. Either the two configurations are equivalent only when n is divisible by six, or the two configurations are equivalent for all but a finite set of values of n. Our guess is that the first possiblity is true.

In this paper we showed how to get the maximum possible score in Blet, but we didn't address the question of speed. How many steps are needed to solve Blet-n? Can some maximal configurations be reached quicker than others? Which configurations (maximal or not) are farthest from the starting configuration?

Another open problem involves the number of possible Blet configurations. We know there are at least $2^{n/2}$ and at most 2^n, and we know the number for some small values of n, but we don't understand this number in general.

Finally, one can play a different game with the Blet rules, starting at a random achievable configuration and trying to go back to $(HT)^{n/2}$.

References

[PV] B. Poonen and F. R. Villegas, *Lattice polygons and the number 12*, Amer. Math. Monthly **107** (2000), no. 3, 238–250.

Department of Mathematics, The University of Texas at Austin, Austin, TX 78712-1082 U.S.A.

E-mail address: sadun@math.utexas.edu

E-mail address: villegas@math.utexas.edu

E-mail address: voloch@math.utexas.edu