Blet: A Mathematical Puzzle

F. Rodriguez Villegas, L. Sadun, and J. F. Voloch

1. THE RULES OF BLET. Blet[1] 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 any 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 also 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 four, six, or eight coins, it is easy to reach the maximum by being greedy, always converting THT to HTH and never converting HTH to THT. With ten coins, you can only get seven heads by being greedy, but there *is* a way to get eight 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 ten-coin puzzle (Blet-10) on your own. (You may prefer using a two-color counter instead of coins. Or you can use pencil and paper. An electronic version, with twenty-eight "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 will 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, kth and $(k+1)$st letters of a word w 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 at the corresponding positions in a word, is a "type-II" move and is denoted II_k. In the electronic version, both are done by clicking on the kth letter, so we refer to either move as "pushing the kth button." Note that pushing the first button changes the first, second, and last letters, whereas pushing the last button changes the second-to-last, last, and first letters. We will say that two words w and w' are *equivalent* if we can obtain one from the other by a succession 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:

$$\begin{array}{rc} HTHHTTTH & \text{(starting configuration)}, \\ THTHTTTH & (II_2), \\ HHTHTTHT & (I_8), \\ HHHTHTHT & (I_4), \\ HHHHTHHT & (I_5). \end{array}$$

[1]The word "Blet" was coined by Malena Villegas. To the best of our understanding, it has no meaning.

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.

With any word w we will associate a path in the space of 2×2 matrices. We start at the identity matrix

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Reading the word w *from right to left*, we move from each matrix M to the matrix $M_H M$ (if the relevant letter is H) or $M_T M$, where

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

Note that M_H and M_T are elements of $SL_2(\mathbb{Z})$, the space of integer matrices with determinant one. As a result, every matrix in the path is likewise in $SL_2(\mathbb{Z})$.

For an arbitrary 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. Moreover,

$$M_H M_T M_H = M_T M_H M_T$$

(check this!), so applying a Blet move to the middle of w will not change $\rho(w)$.

For the example $w = HTHHTTTH$ in (1), we have:

$$\begin{aligned}
M_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\
M_1 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = M_H, \\
M_2 &= \begin{pmatrix} 1 & 1 \\ -1 & 0 \end{pmatrix} = M_T M_H, \\
M_3 &= \begin{pmatrix} 1 & 1 \\ -2 & -1 \end{pmatrix} = M_T M_T M_H, \\
M_4 &= \begin{pmatrix} 1 & 1 \\ -3 & -2 \end{pmatrix} = M_T M_T M_T M_H, \\
M_5 &= \begin{pmatrix} -2 & -1 \\ -3 & -2 \end{pmatrix} = M_H M_T M_T M_T M_H, \\
M_6 &= \begin{pmatrix} -5 & -3 \\ -3 & -2 \end{pmatrix} = M_H M_H M_T M_T M_T M_H, \\
M_7 &= \begin{pmatrix} -5 & -3 \\ 2 & 1 \end{pmatrix} = M_T M_H M_H M_T M_T M_T M_H, \\
M_8 &= \begin{pmatrix} -3 & -2 \\ 2 & 1 \end{pmatrix} = M_H M_T M_H M_H M_T M_T M_T M_H = \rho(w).
\end{aligned} \tag{2}$$

To understand the path in $SL_2(\mathbb{Z})$, we consider separately the rows of the state matrix M. Let q denote its first row and p its second row. In terms of p and q, the actions of H and T are:

$$H : \begin{cases} q \mapsto q + p \\ p \mapsto p \end{cases}, \qquad T : \begin{cases} q \mapsto q \\ p \mapsto p - q \end{cases}.$$

Note that H does not change p, while T does not change q.

By joining each vector q to the next, we obtain a polygonal path in the plane that we will denote by $\mathcal{Q}(w)$. Likewise, by tracking the second row, we obtain a polygonal path $\mathcal{P}(w)$. (A program that draws the pictures given a word is available at http://www.ma.utexas.edu/users/villegas/nmx.html .) For our running example (1) these look as follows.

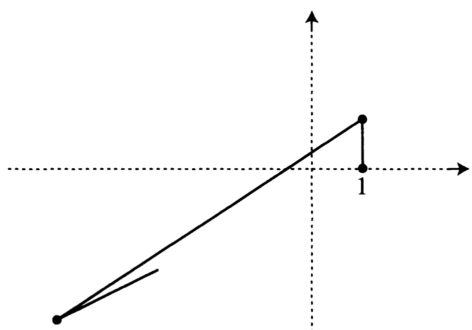

Figure 1. The path $\mathcal{Q}(HTHHTTTH)$.

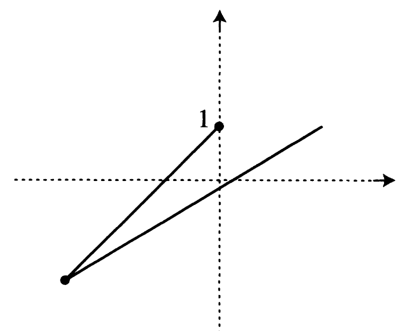

Figure 2. The path $\mathcal{P}(HTHHTTTH)$.

You may have noticed that not every q-vector in (2) is a vertex of $\mathcal{Q}(w)$. When there are several Hs in succession, q moves in a straight line, in the p-direction. The edges of $\mathcal{Q}(w)$ therefore correspond to runs of one or more consecutive Hs in w, while the vertices of $\mathcal{Q}(w)$, where the direction changes, correspond to runs of one or more consecutive Ts.

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 $\rho(w)$ being the identity matrix. You should check that $HTHTHTHTHTHT$ is closed, and that every starting Blet configuration is eventually closed.

Our first goal is to prove the following.

Theorem 1. *If w is an eventually closed word, then*

$$\frac{\ell}{6} < \ell_H < \frac{5\ell}{6}, \qquad \frac{\ell}{6} < \ell_T < \frac{5\ell}{6}. \tag{3}$$

Proof. In order to establish (3) we will relate the lengths ℓ_H, ℓ_T, and ℓ of a word to geometric data about the path $\mathcal{Q}$. First we relate ℓ to the winding number of $\mathcal{Q}$, using a formula that was proved in [7] 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.

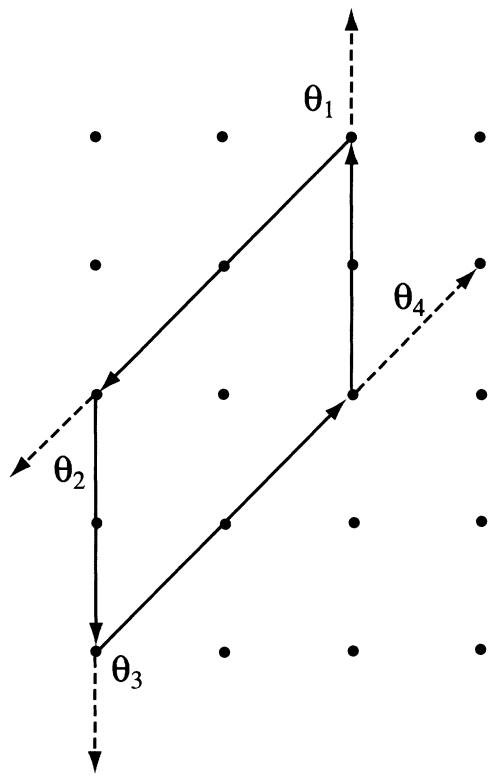

Figure 3. Exterior angles add up to $2\pi m(\mathcal{Q})$.

Lemma 2 (Poonen and Rodriguez Villegas). *Let w be a closed word of total length ℓ, and let $\mathcal{Q} = \mathcal{Q}(w)$ be its associated path. Then*

$$\ell = 12m(\mathcal{Q}). \tag{4}$$

Now suppose that w is closed. We consider the vertices $v_1, \ldots, v_r$ 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).

It is not hard to see that

$$\sum_{j=1}^{r} \theta_j = 2\pi m(\mathcal{Q}). \tag{5}$$

Combining (5) with (4), we obtain

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

Since each exterior angle θ_j is strictly less than π, the number r of such angles is strictly greater than $\ell/6$. Because there are one or more Ts associated to each vertex, the number of Ts is greater than $\ell/6$. Likewise, the number of Hs is at least the number of edges, which is the same as the number of vertices and is thus greater than $\ell/6$.

So far we have assumed that w is closed. If w is merely eventually closed, then w^n is closed for some n. By the above arguments, the fraction of Hs in w^n is strictly between 1/6 and 5/6. Since the fraction of Hs in w is the same as in w^n, the theorem is proved. ∎

3. OPTIMAL BLET CONFIGURATIONS. 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 $\rho(w)$ is the identity. Since $M_H M_T M_H = M_T M_H M_T$, pressing the second, third, ..., or second-to-last button does not 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 $\rho(w')^k$ is the identity if and only if $\rho(w)^k$ is. As a result, all legal Blet configurations are eventually closed, and we have proved:

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 is easy to see that $b(2) = 1$, $b(4) = 3$, and $b(6) = 4$.

Theorem 4. *The inequality $b(n + 6) \geq b(n) + 5$ holds for $n = 2, 4, \ldots$.*

We will prove this theorem shortly, but first let us 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 the same as the upper bounds given by Theorem 3. Thus the upper bounds are achievable—and 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.*

Proof of Theorem 4. We view the $(n + 6)$-letter starting configuration as an n-letter "body" and a six-lettter "tail" $HTHTHT$. By assumption, we can convert the body into a word with $b(n)$ Hs and $n - b(n)$ Ts, 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 seven letters to $HTHHHTH$. There are now $b(n) + 5$ heads, $b(n)$ in the first $n - 1$ letters and five in the last seven 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:

$$\begin{aligned} H(HTHTHT) &= HHTHTHT \to HTHTTHT \to HTHTHTH \\ &= (HTHTHT)H. \end{aligned} \tag{6}$$

We call the combination of moves in (6) a *slide*. Notice that the slide does not change the total number of Hs—it just makes an $HTHTHT$ unit swap places with an H.

If the body ends with a T followed by k Hs, 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:

$$
\begin{array}{ll}
HTHT\ HTHTHT & \text{(starting position with 4-letter body} \\
 & \quad \text{and 6-letter tail),} \\
HHTH\ HTHTHT & (I_3 \text{ acts on body}), \\
HHTHTHTHTH & (\text{slide } H \text{ past tail; } II_6 \text{ and } I_9), \\
HHHTHHHTHH & (I_4 \text{ and } I_8).
\end{array}
\tag{7}
$$

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

$$
\begin{array}{ll}
HTHTHTHTHT\ HTHTHT & \text{(start with 10-letter body} \\
 & \quad \text{and 6-letter tail),} \\
HHHTHHHTHHHTHTHT & \text{(manipulate body as in (7)),} \\
HHHTHHHTHHTHTHTH & (\text{slide one } H \text{ past tail}), \\
HHHTHHHTHTHTHTHH & (\text{slide another } H \text{ past tail}), \\
HHHTHHHHHTHHHTHHH & \text{(final two type-I moves).}
\end{array}
$$

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 a 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 (five possible configurations) and Blet-6 (eight configurations), but the system gets more complicated quickly. Blet-28 has over eleven million possible configurations, of which only 196 have the maximum (twenty-three) 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 does not work in this instance!). Starting from the initial position, make a sequence of moves such that each move increases the number of Hs until it is impossible to do so. For Blet with twenty-eight coins, the greedy algorithm will never go beyond twenty-one heads.

In the simplified form of simulated annealing we first choose a number ϵ satisfying $0 < \epsilon < 1$. We then make a sequence of moves, mostly trying to increase the number of Hs but allowing moves that decrease the number of Hs "ϵ fraction of the time." More precisely, starting from our initial position, we make a sequence of moves as follows. From the current position, select a random valid move and a random number δ, where $0 < \delta < 1$, in a process akin to tossing a die. We make the selected valid move either if the move will increase the score or if $\delta < \epsilon$. As described, this procedure may go on forever. It may also reach a maximum and subsequently leave it. Therefore, it is necessary to keep track of the largest score encountered and put a limit on the number of iterations. This algorithm is only practical in certain circumstances, but it does work very well for Blet.

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 115,929 local maxima out of 11,698,223 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 a player.

We should mention that in more sophisticated versions of simulated annealing the ϵ may depend on the score and on the number of iterations. This process is motivated by the metallurgical procedure of annealing, in which a metal is initially heated and then left to cool slowly so as to achieve a low-energy position. The parameter ϵ corresponds to the temperature of the metal, with the greedy algorithm $\epsilon = 0$ corresponding to freezing the system at absolute zero temperature. The score corresponds to the energy of the metal, and the number of iterations represents time. For more about simulated annealing, see **[6]** for the original reference, **[3]** for an interesting application, **[1]** for general theory, and **[5]** for an analysis of the efficiency of holding ϵ fixed.

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 us 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 whose edges link positions that are one move apart. We proceed to show that the number of vertices grows exponentially. Let B_n denote the number of configurations of Blet-n.

Theorem 6. *For $n > 2$ it is true that $2^{n/2} \leq B_n \leq 2^n$.*

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 modeling 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 six 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 six 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: $HTHTHT$, $THTTHT$, $HHTHHT$, $HTTHTT$, $THTHTH$, $HTHHTH$, $TTHTTH$, $THHTHH$. 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 do not know the exact number of vertices in general, but here is some data for small n:

TABLE 1.

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 to our lower bound.

The data in Table 1 was obtained by constructing a spanning tree for the graph of Blet. A *tree* is a connected graph with no circuits. A *spanning tree* of a graph is a subgraph that contains all vertices of the graph and, moreover, is a tree. We used a standard spanning tree algorithm, called *depth-first*, which goes as follows. We start with the initial position and keep track of all visited positions, together with the moves that first brought us to them. If we are in a certain position, we look for a move that will take us to a new position. If one exists, we move to it and add the new position to our set of visited positions; otherwise we backtrack from our current position using the recorded move that first brought us there, unless the position is the initial one. In the case of the initial position, if we return to it and cannot move to a new position, we terminate the algorithm. The reader can check that this algorithm always terminates in a spanning tree. The referee suggested dynamic programming as an alternative approach. A suitable reference for these various algorithms is [**4**].

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 four Hs, a T, six Hs, a T, and three Hs. In general, we will denote by $(s_1, \ldots, s_k)$ any configuration that is a cyclic permutation of $TH^{s_1}TH^{s_2}\cdots TH^{s_k}$. The configuration $(4, 6, 3)$ 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 6,885 possible Blet-16 configurations, only sixteen are global maxima.

From the proofs of Theorems 4 and 6, we obtain a procedure that yields 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 $(s_1, \ldots, s_k)$ has been obtained for Blet-n, we 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 pattern to $HTHHHTH$. 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 proved:

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

We can actually say a little more.

Theorem 8. *Every optimal Blet configuration is obtainable recursively by the procedure indicated in Theorem 7.*

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 achieves equality in the upper bounds of Theorem 3 is obtained in this way. As a corollary, this shows that every such word is equivalent either to $(HT)^{n/2}$ 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_is are equal to one or two; (3) at least one of the s_is 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 gives 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 one, 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 be one. Now suppose that $s_i = 2$ and $n \geq 12$, so w contains at least three Ts. If $s_i = 2$, there exists the pattern HTHHTH somewhere in w. 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

$$\rho(TH^2TH^6) = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}$$

is the identity, so words of the form (2, 6) are not eventually closed.

Step 3 requires a lemma:

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

Proof. We explicitly compute

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

where

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

These matrices satisfy the relations

$$Y^2 = 0, \quad X^2 = X, \quad XYX = X, \quad YXY = Y.$$

If all the s_is are equal to four, then we have $\rho((TH^4)^k) = (-1)^k(I + kY)$, which is not the identity. If at least one of the s_is is greater than four, but none is less than four, 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 $(s_1, \ldots, s_k)$ 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 $(s_1, \ldots, s_k)$ with each of the s_is at least three, and at least one of the s_is equal to three. Without loss of generality, we can assume that w is of the form $(s_1, 3, s_3, \ldots, s_k)$. The word 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 $n - 6$ gotten by replacing $THTHTHT$ with T is eventually closed, and it has five fewer Hs than the 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 can 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 that 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. FURTHER PROBLEMS. First, as a warm-up, the reader might try the following problem. It is easy to see that $(HT)^{n/2}$ and $(TH)^{n/2}$ are equivalent when n is divisible by six—just push every third button. Prove that, conversely, these two configurations are equivalent only when n is divisible by six.

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? We don't know the answers to these questions.

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 precise 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 $(HT)^{n/2}$ (rather like solving a Rubik's cube).

In considering some of these questions the connection, which we have not yet mentioned, between the basic relation $HTH = THT$ and the braid group on three strands is likely to be useful (see, for example, Burckel [**2**]).

ACKNOWLEDGMENTS. We would like to thank R. Chapman, N. Elkies, H. Thomas, D. Zagier, and the referee for various suggestions on an earlier draft of this paper, our children for trying out the game and, especially, Malena Villegas for suggesting the name Blet.

REFERENCES

1. E. H. L. Aarts and P. J. M. van Laarhoven, *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, The Netherlands, 1987.
2. S. Burckel, Syntactical methods for braids of three strands, *J. Symbolic Comput.* **31** (2001) 557–564.
3. S. Carlson, Algorithm of the gods, *Scientific American* (March 1997) 121–123.
4. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, 1990.
5. M. Fielding, Simulated annealing with an optimal fixed temperature, *SIAM J. Optim.* **11** (2000) 289–307.
6. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science* **220** (4598) (1983) 671–680.
7. B. Poonen and F. Rodriguez Villegas, Lattice polygons and the number 12, *Amer. Math. Monthly* **107** (2000) 238–250.

FERNANDO RODRIGUEZ VILLEGAS obtained the degree of Licenciado en Ciencias Matemáticas from the Universidad de Buenos Aires, Argentina, in 1985 and received a Ph.D. in mathematics from Ohio State University in 1990. Except for a year at the Institute for Advanced Study (1990–91) and a year at the Max Planck Institut in Bonn (1994–95) he was at Princeton University before joining the faculty at the University of Texas in 1998. He spent the year 2001–02 at Harvard University. His main research interests are in number theory, modular forms, and special values of L-functions. He is currently an Alfred P. Sloan Research Fellow.
University of Texas at Austin, TX 78712-1082
villegas@math.utexas.edu

LORENZO SADUN obtained a Ph.D. in physics from the University of California at Berkeley in 1987, while studying at the Harvard Math Department. After postdoctoral positions at Caltech and New York University, he joined the math department of the University of Texas in 1991. His research is frequently collaborative, and

covers diverse problems in geometry, quantum mechanics, and their interface. His coauthors to date include twenty-three mathematicians, eight physicists, a biologist, a computer scientist, and a stuffed dragon.
University of Texas at Austin, TX 78712-1082
sadun@math.utexas.edu

JOSÉ FELIPE VOLOCH grew up in Rio de Janeiro, Brazil. He received his Ph.D. degree in mathematics from the University of Cambridge (UK) in 1985. He then returned to his hometown, working at IMPA until 1992. After a year at UC Berkeley, he took a position at the University of Texas, where he has been for ten years. His research interests include number theory, algebraic geometry, and coding theory. He likes to program and play with computers in his spare time.
University of Texas at Austin, TX 78712-1082
voloch@math.utexas.edu