Hat Guessing Games

*Steve Butler** *Mohammad T. Hajiaghayi*† *Robert D. Kleinberg*†‡§

Tom Leighton†¶

Abstract

Hat problems have become a popular topic in recreational mathematics. In a typical hat problem, each of n players tries to guess the color of the hat they are wearing by looking at the colors of the hats worn by some of the other players. In this paper we consider several variants of the problem, united by the common theme that the guessing strategies are required to be deterministic and the objective is to maximize the number of correct answers in the worst case. We also summarize what is currently known about the worst-case analysis of deterministic hat-guessing problems with a finite number of players.

Key words: hat game; deterministic strategies; sight graph; Tutte-Berge formula; hypercube

AMS Mathematics Subject Classification: 91A12; 05C20

1 Introduction

Consider the following game. There are n distinguishable players and one adversary. The adversary will place on the heads of the players hats of k different colors, at which point players are allowed to see all hats but their own. No communication is allowed. Each player then makes a private guess as to what hat they are wearing. The goal of the players is to maximize the number of correct guesses.

To help players maximize their correct guesses, the players are allowed to meet before the hats are placed on their heads and to determine a public deterministic strategy (public in that everyone, including the adversary, knows the strategy and deterministic in that the guesses are determined

*Department of Mathematics, University of California, San Diego, La Jolla, CA 92093, U.S.A., Email: `sbutler@math.ucsd.edu`.

†Department of Mathematics and Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139, U.S.A., Emails: {`hajiagha,rdk,ftl`}`@theory.csail.mit.edu`.

‡Computer Science Division, UC Berkeley, Berkeley, CA 94720; and Department of Computer Science, Cornell University, Ithaca, NY 14853.

§Part of this work was supported by a National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship.

¶Akamai Technologies, Eight Cambridge Center, Cambridge, MA 02139, U.S.A.

completely by the hat placement). What is the maximum number of correct guesses that can be guaranteed, and what strategy should be implemented to achieve the maximum?

This question has been answered in [3, 9]. Here we will present the answer and consider some variations on the game, showing that they lead to some surprisingly subtle combinatorial and algorithmic problems and theorems. We proceed as follows. In the remainder of the introduction we answer these questions for the game as stated. In Section 2 we consider what happens when players are not able to see everyone, which might be applicable as an abstraction of the problem of guessing information globally in a market or a decentralized computing system, in which each person (or software agent) has only partial/local knowledge. In Section 3 we give a hypercube interpretation of the game which gives insight into the nature of optimal strategies, and we explore a version of the hats game where the adversary has a restricted hat supply.

Hat guessing games have long been a popular source of problems in recreational mathematics, and variations of hat guessing games have recently attracted increasing attention [6], partly because of their connections with coding theory (particularly Hamming codes). As another example, Aggarwal et al. [1] have used hat problems in the design of deterministic auction mechanisms. When constructing truthful auction mechanisms, a mechanism designer must devise a procedure for assigning a price to each bidder based only on the bids of other players (else she may have an incentive to lie about her bid), with the aim of charging many bidders a price which is close to their own bid. Note the formal similarity with hat problems, in which the goal is to devise a procedure for assigning a guess to each player based only on the hat colors of other players, with the aim of assigning to many players a guess which matches their own hat color. By exploiting this similarity between the two problems, Aggarwal et al. used hat-guessing strategies for a variant of the balanced hat problem (discussed below in Section 3.1) to provide a generic procedure for converting a randomized auction into a deterministic auction with approximately the same revenue, in markets with single-parameter bidders and no supply constraints.

Finally it is worth mentioning the fact that we are considering deterministic strategies instead of randomized ones is very important in this paper. The main reason is that these deterministic strategies focus on the worst-case scenarios instead of average or even almost-all scenarios. As a result, for several problems in this paper obtaining a randomized algorithm which guesses a constant fraction of the desired hat colors on average is easy though we cannot even guess one or a constant number of hat colors deterministically (see Sections 2.3 and 2.4).

1.1 A winning approach to the hat guessing game

Example 1. Consider the case where there are 2 players and 2 colors of hats. Then a winning strategy for these players is to have the first player guess what the second player is wearing and the second player guess the color opposite of what the first player is wearing. If they are wearing hats of the same color then the first player guesses correctly. If they are wearing different colors then the second player guesses correctly. In any case there is one correct guess (and one incorrect guess). □

It is interesting to note that in this example the expected number of correct guesses is 1, the same as if they had guessed randomly. What their strategy has done is to eliminate the variance involved in the guessing. The other thing to note is that neither player has any idea who guessed correctly, but they do know that collectively one of them did. These two properties will hold in general.

We have the following general result, first proved for 2 colors by Winkler [9] and later generalized to k colors by Feige [3].

Theorem 2. *If there are n players and hats of k different colors then there exists a strategy guaranteeing at least $\lfloor n/k \rfloor$ correct guesses. No strategy can improve on this.*

Proof. We first demonstrate a strategy. Number the players 1 to n and the colors of the hats 1 to k. The ith player will guess as if the sum of all the hats (including their own) is congruent to i mod k. At least $\lfloor n/k \rfloor$ of the players will be acting correctly and will therefore guess correctly.

To see that this cannot be improved upon we use an averaging argument. If a player sees a particular placement of hats then they are in one of k situations and they will guess correctly in exactly one of these situations. Since there are k^{n-1} ways to place the hats on the remaining players we see that each player will make k^{n-1} correct guesses over all possible placements of hats. Since there are n players and k^n ways to place the hats then on average we have $nk^{n-1}/k^n = n/k$ correct guesses. It follows that the adversary can find some placement of hats with at most $\lfloor n/k \rfloor$ correct guesses. $\square$

2 Restricting our vision in the game

In the original version of the game every player can see every other player. In an actual implementation of the game with a large group of people this might be difficult to achieve. So we now consider a variation where each player sees some subset of the other players.

To do this we introduce another layer to the game. We consider the "sight graph" where the vertices are the players and we have a directed edge from $a \to b$ if player a can see player b. As an example, in the original version of the game the graph was the complete graph on n vertices. For a given sight graph G we will let $H(G)$ denote the maximum number of correct guesses that the players can guarantee using an optimal strategy when there are 2 colors of hats.

In this section we will first consider the undirected case, i.e., the case in which every directed edge (u, v) is accompanied by the reverse edge (v, u). For this case, an exact answer to $H(G)$ is known. In the directed case no exact answer for $H(G)$ is known but simple lower and upper bounds do exist. Finally, we consider the case when there are more than 2 colors of hats, for which little is known.

2.1 The undirected case

When we have an undirected graph the obvious strategy is to have players pair up as best as possible. Then in each pair we can implement the strategy in Example 1. This shows that we can guarantee at least $|M|$ correct guesses where M is a maximum matching of G. The next result shows that this cannot be improved upon.

Theorem 3. *Let G be an undirected graph with M a maximum matching of G. Then $H(G) = |M|$.*

Proof. It remains to show that $H(G) \le |M|$. To do this we use the Tutte-Berge formula [2, 8], which says that there is a subset U of the vertices such that

$$|M| = \frac{|V| + |U| - o(G-U)}{2},$$

where $o(G-U)$ is the number of connected components of the induced subgraph $G-U$ which have an odd number of vertices. For $j = o(G-U)$ let $W_1, \ldots, W_j$ be the connected components of $G-U$ which have an odd number of vertices and Y the union of all the connected components of $G-U$ which have an even number of vertices.

Given any strategy we place hats as follows. First place hats on U arbitrarily. Having fixed the hat placement on U, for each player in W_i their guess is now completely determined by the hat placement on W_i (since the only other players that can be seen are U which has already been placed). Applying the arguments from Theorem 2 there is some placement of hats on each W_i with at most $(|W_i| - 1)/2$ correct guesses. Similarly we can place hats on Y so that there are at most $|Y|/2$ correct guesses. Therefore the total number of correct guesses is bounded above by

$$|U| + \frac{|Y|}{2} + \frac{|W_1| - 1}{2} + \cdots + \frac{|W_j| - 1}{2} = \frac{|V| + |U| - j}{2} = |M|. \qquad \square$$

2.2 The directed case

For the directed case there is no obvious strategy to adopt, and no sharp bound for $H(G)$ is known. However there exist simple upper and lower bounds as shown in the following.

Lemma 4. *Given a directed graph G let $c(G)$ denote the maximal number of vertex disjoint cycles in G, and $F(G)$ denote the minimum number of vertices whose removal from G makes the graph acyclic. Then $c(G) \le H(G) \le F(G)$.*

Proof. The lower bound follows by noting that for every cycle we can guarantee one correct guess. For example, if we have a cycle $a_1 \to a_2 \to \cdots \to a_k \to a_1$ then by having players $a_1, \ldots, a_{k-1}$ guess the opposite color of the next player and a_k guess the color of the hat a_1 has, we guarantee at least one correct guess.

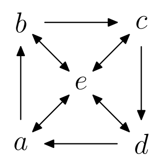
For the upper bound we note we can arrange the vertices in order so that the removal of $v_1, \ldots, v_{F(G)}$ leaves the graph acyclic and the remaining vertices $v_{F(G)+1}, \ldots, v_n$ are such that if $i > F(G)$ and there is an edge from v_i to v_j then $j < i$. In other words for the last $n - F(G)$ vertices, all outgoing edges point to the left. We place hats on the first $F(G)$ players arbitrarily and then we can place hats on players $F(G)+1$ to n in turn, choosing each of the last $n-F(G)$ hat colors so as to force the corresponding player to guess incorrectly, given the colors of the preceding players. $\square$

By a theorem of Reed et al. [4], formerly known as Younger's Conjecture (namely, for every integer $k \ge 0$ there exists an integer $t \ge 0$ such that every digraph G has k vertex disjoint directed cycles, or G can be made acyclic by deleting at most t vertices), this implies a criterion for determining whether a family of directed graphs has unbounded "hat number."

Corollary 5. *Let $\mathcal{G}$ be a set of finite directed graphs. The set $\{H(G) : G \in \mathcal{G}\}$ is unbounded if and only if the set $\{F(G) : G \in \mathcal{G}\}$ is unbounded.*

Neither bound in Lemma 4 is sharp. For the upper bound, the undirected triangle has $F(G) = 2$ but we know from Theorem 2 that $H(G) = 1$. An example to show the lower bound is not sharp is a little more involved and is given below.

Example 6. Let G consist of a directed four cycle $a \to b \to c \to d \to a$ with a fifth node e joined to the other four by bi-directed edges. This graph has $c(G) = 1$ and $H(G) = 2$.



To describe a strategy we will let A, B, C, D, E denote the actual colors of hats placed on players a, b, c, d, e respectively, while g_a, g_b, g_c, g_d, g_e denote their guesses. We can describe their strategy in mod 2 arithmetic as follows.

$$g_a = B + E; \qquad g_b = C + E; \qquad g_c = D + E; \qquad g_d = A + E + 1;$$

$$g_e = \begin{cases} 1 & \text{if } (A+B, B+C, C+D, A+D+1) \text{ has Hamming weight 1;} \\ 0 & \text{if } (A+B, B+C, C+D, A+D+1) \text{ has Hamming weight 3.} \end{cases}$$

What happens is that the players a, b, c, d will make either 1 or 3 correct guesses depending on what e is wearing. So e guesses as though his/her hat would force 1 correct guess among the other four players. Thus, either e guesses wrong and there are 3 correct guesses among a, b, c, d; or e guesses correctly and there is 1 correct guess among a, b, c, d for a total of 2 correct guesses. $\square$

Question. For an undirected graph G how do we calculate $H(G)$? The obvious algorithm for deciding if $H(G) \geq h$ requires nondeterministic exponential time: the algorithm nondeterministically comes up with a guessing strategy for the players, and then spends exponential time verifying that this strategy produces at least h correct answers on every input. We do not know if there is a more efficient algorithm for deciding if $H(G) \geq h$.

We note that in answering this question for directed graphs it suffices to consider graphs G which are strongly connected. In particular, $H(G) = \sum_k H(G_k)$ where G_k are the strongly connected components of G. The proof of this is similar to the argument of the upper bound in Lemma 4. Namely, we can order the strongly connected components so that there is an edge from G_i to G_j only if $i > j$. Then the adversary acts optimally on each of the connected components in turn.

A related question concerns the complexity of optimal guessing strategies. Define a guessing strategy with sight graph G to be *optimal* if it achieves at least $H(G)$ correct answers on every input. We saw in the proof of Theorem 3 that when G is undirected, there is always an optimal guessing strategy in which each player's guess is computed by evaluating a linear function (over the field with two elements) whose inputs are the other players' hat colors together with the constant 1. For directed graphs this is not the case. The sight graph in Example 6 has $H(G) = 2$, but the reader may verify by a simple case analysis that for every linear guessing strategy, there exists an input on which fewer than 2 players answer correctly.

Question. Is it true that for every directed graph G, there is an optimal guessing strategy in which every player's guess is computed by inserting the other players' hat colors into a Boolean circuit of size polynomial in $|G|$?

Note that an affirmative answer to this question would imply that the problem of deciding if $H(G) \geq h$ belongs to the complexity class Σ_2^P, providing a partial answer to the preceding question.

2.3 More than 2 colors of hats

Considerably less is known when there are more than 2 colors involved in the game. Let $H_k(G)$ denote the maximum number of correct guesses that the players can guarantee using an optimal strategy when there are k colors of hats. From the proof of Theorem 2, we know that $H_k(G) = 1$ when G is an undirected k-clique, and therefore $H_k(G) \geq \ell$ whenever G contains ℓ disjoint undirected k-cliques. However it is possible to avoid k-cliques altogether and still guarantee at least one correct guess, as shown below.

Theorem 7. *For every number k, there exists a bipartite graph G with $H_k(G) > 0$.*

Proof. Let G be a complete bipartite graph with $n = k - 1$ vertices on the left side and $m = k^{k^n}$ vertices on the right side. Let C denote the set of all k-colorings of the left side of G. Note that $|C| = k^n$ and $m = k^{|C|}$, hence m is equal to the number of mappings from C to $\{1, 2, \ldots, k\}$. Pick a one-to-one correspondence between the vertices on the right side of G and the mappings from C to $\{1, 2, \ldots, k\}$, and let each vertex on the right side of G guess its color using the corresponding mapping.

We will need the following claim.

Claim. Let c_R denote a fixed coloring of the right side of G, and let C' denote the set of all colorings c_L of the left side of G such that the combined coloring (c_L, c_R) causes every vertex on the right side to guess its color incorrectly. Then $|C'| < k$.

Now it's time to define the guessing strategies used by the vertices on the left side of G. Given the coloring of the right side, the set C' defined in the lemma above has at most $n = k-1$ elements. So let $c_1, c_2, \ldots, c_n$ be a list of colorings which contains every element of C'. For $i = 1, 2, \ldots, n$, vertex i on the left guesses that its color is $c_i(i)$. This guessing strategy (combined with the guessing strategy for the vertices on the right side as defined above) guarantees at least one correct answer. This is because the above claim guarantees that at least one vertex on the right side guesses correctly unless the coloring of the left side belongs to C'. But if the coloring of the left side belongs to C', then it is equal to c_i for some i in $1, 2, \ldots, n$, in which case vertex i on the left guesses its color correctly.

It remains to prove the claim. The proof follows from noting that if C' contains k distinct elements $c_1, c_2, \ldots, c_k$, then there exists a function f from C to $\{1, 2, \ldots, k\}$ which assumes k distinct values on the set $\{c_1, \ldots, c_k\}$. Let v denote the vertex on the right side of G corresponding to f. Since the set $\{f(c_1), f(c_2), \ldots, f(c_k)\}$ contains all k colors, we must have $f(c_i) = c_R(v)$ for some i in $1, 2, \ldots, k$. Thus, the combined coloring (c_i, c_R) causes vertex v to guess its color correctly, contradicting our assumption that c_i belongs to C', ending the proof. $\square$

Question. Is there a bipartite graph G satisfying $H_k(G) > 0$ whose size is polynomial in k? What if instead of bipartite we consider k-clique-free graphs?

Question. Call an undirected sight graph G "edge-critical for the hats game with k colors" if G has the property that there exists a guessing strategy which guarantees at least one correct answer for the hats game with k colors, but no proper subgraph of G has this property. For $k = 2$, the only edge-critical graph is a 2-clique. For $k > 2$, there are at least two (undirected) edge-critical graphs, namely a k-clique and a subgraph of the complete bipartite $(k-1)$-by-$(k^{k^{k-1}})$ graph. For $k > 2$, are there infinitely many graphs which are edge-critical for the hats game with k colors?

We close this section by proving that $H_k(G) = 0$ whenever $k > 2$ and G is an undirected tree. This fact is a consequence of the following more general lemma.

Lemma 8. *Suppose we are given: an undirected tree T; a guessing strategy Γ for the hat k-coloring problem on T; a node v in T; and a pair of colors c_1, c_2. Then there exists a k-coloring ($k \geq 3$) of T such that every node guesses its color incorrectly; and the color of node v is either c_1 or c_2.*

Proof. The proof is by induction on the size of T. When $|V(T)| = 1$ the result is trivial. Otherwise deleting v from T partitions the remaining vertices into a collection of disjoint subtrees T_1, T_2, …, T_j. For $i = 1, 2, \ldots, j$, let $r(T_i)$ denote the unique neighbor of v in T_i. Let $\Gamma_1(T_i)$ (respectively $\Gamma_2(T_i)$) denote the guessing strategy applied in T_i when the color of v is c_1 (respectively c_2). Note that $\Gamma_1(T_i)$ and $\Gamma_2(T_i)$ differ only in the function which $r(T_i)$ uses to guess its color based on the colors of its neighbors in T_i. Let $B_1(T_i)$ denote the set of "bad colorings" for guessing strategy $\Gamma_1(T_i)$, i.e., the colorings which cause every node of T_i to guess its color incorrectly. Let $C_1(T_i)$ denote the set of colors assigned to $r(T_i)$ by colorings in $B_1(T_i)$. Define sets $B_2(T_i)$, $C_2(T_i)$ similarly, but using the guessing strategy $\Gamma_2(T_i)$ in place of $\Gamma_1(T_i)$. The induction hypothesis implies that $C_1(T_i)$ and $C_2(T_i)$ each have at least $k - 1$ elements. (If not, then the complement of one of these sets, say $C_1(T_i)$, contains at least two colors, say c_3, c_4. Applying the induction hypothesis with tree T_i, guessing strategy $\Gamma_1(T_i)$, node $r(T_i)$, and color pair c_3, c_4 would lead to an element of $B_1(T_i)$ in which the color of $r(T_i)$ is either c_3 or c_4, contradicting the assumption that c_3, c_4 are both in the complement of $C_1(T_i)$.) Having established that $C_1(T_i)$ and $C_2(T_i)$ each have at least $k - 1$ elements, it follows (from the fact that $k > 2$) that the intersection of $C_1(T_i)$ and $C_2(T_i)$ is non-empty. Choose a color c_i from the intersection of these two sets and assign it to $r(T_i)$. Do this for each i in $\{1, 2, \ldots, j\}$. Having assigned a color to each neighbor of v, the guess of node v is now determined. At least one of the colors c_1, c_2, differs from this guess, so we may assign this color to node v and thereby ensure that it guesses incorrectly. Assume without loss of generality that color c_1 is assigned to v. For each subtree T_i, the set $B_1(T_i)$ contains a coloring which satisfies:

- the color of $r(T_i)$ is c_i;
- every node guesses its color incorrectly using guessing strategy $\Gamma_1(T_i)$.

We choose one such coloring and use it to assign colors to the nodes of T_i. Doing this for every i in $\{1, 2, \ldots, j\}$ yields a coloring of T which satisfies the two properties in the statement of the lemma. $\square$

Corollary 9. *If G is an undirected tree and $k > 2$ then $H_k(G) = 0$.*

2.4 Generalized guessing graphs

In this section we consider a variation in which players are not necessarily trying to guess their own hat color. Instead there is a set P ("players") and a set H ("hats"), and two directed graphs G_v ("visibility graph") and G_g ("guessing graph"). Both graphs have a vertex set which is the union of P and H. Every edge of G_v has its tail in P and its head in H; we think of edge (u, v) as indicating that person u can see the color of hat v. Every edge of G_g has its tail in H and its head in P; we think of edge (v, u) as indicating that person u must guess the color of hat v. (Note that the orientation of these edges is from hats to people, the reverse of the orientation convention used in G_v. This orientation convention is adopted because it will be convenient later on.) The hat problem considered in earlier sections corresponds to the case when there is a bijection $\phi : H \to P$ and the edge set of G_g is $\{(v, \phi(v)) : v \in H\}$.

A "guessing strategy" is a set of functions, one for each edge in G_g. Each such function maps the set of k-colorings of H to the set of colors, and has the property that the value of the function corresponding to edge $e = (v, u)$ depends only on the colors of the elements of H which are adjacent to u in G_v. Given a k-coloring of H and a guessing strategy, we say that edge $e = (v, u)$ of G_g gives a correct answer if its function evaluates to the color which was assigned to v. We define $H_k(G_v, G_g)$ to be the maximum number of correct guesses that the players can guarantee using an optimal strategy when there are k colors of hats.

Theorem 10. *When $k = 2$, $H_k(G_v, G_g) > 0$ if and only if at least one of the following properties holds:*

a. There is a vertex of G_g whose outdegree is greater than 1.

b. There is a directed cycle in the union of G_v and G_g.

Proof. Identify the set of colors with the set $\{0, 1\}$. If property (a) is satisfied and G_g contains edges (v, u) and (v, u') for some v in H and u, u' in P, then assign the constant function 0 to edge (v, u) and the constant function 1 to edge (v, u'). Clearly, on every input, at least one of these edges gives a correct answer.

If property (b) is satisfied, let the vertices of the cycle be

$$v_1 \to u_1 \to v_2 \to u_2 \to \cdots \to v_n \to u_n$$

and adopt the following guessing strategy. For $i = 1, 2, \ldots, n-1$, player u_i guesses that the color of v_i is different from the color of v_{i+1}. Player u_n guesses that the color of v_n is the same as the color of v_1. Observe that this is a legal guessing strategy since each of the edges $(u_1, v_2), (u_2, v_3), \ldots, (u_n, v_1)$

belongs to G_v. Also, for any input on which none of $u_1, u_2, \ldots, u_{n-1}$ guess correctly, it must be the case that $v_1, v_2, \ldots, v_n$ are all assigned the same color. But then (u_n, v_1) guesses correctly.

Finally, suppose neither (a) nor (b) is satisfied; we must prove that for every guessing strategy there exists an input on which every edge guesses incorrectly. We will only sketch this part of the proof. Let G be the union of G_v and G_g, and let G' be the directed graph obtained from G by contracting the edges of G_g. If G' contains a directed cycle, then G must also contain a directed cycle. (In fact, our assumption that property (a) is not satisfied implies that every edge of G' corresponds to a 2-hop path between two elements of P in G.) Since we are assuming G contains no directed cycles, it follows that G' is acyclic. An elementary induction argument, using a topological sort of G', produces a coloring of H which causes every edge to guess incorrectly. □

Question. Above we characterize visibility and guessing graphs for which we can guarantee at least one correct answer. It would be nice if we can determine exactly how much more information in the guessing graph we can obtain by adding a particular edge to the visibility graph. More generally, given m, G_v and G_g determine the smallest value j such that there exists a graph G'_v consisting of G_v with j additional edges such that the hat number $H_2(G'_v, G_g)$ is at least m?

The above question can be loosely considered in the same line as the Aanderaa-Rosenberg Conjecture [7] which asks the minimum number of edges of a graph that should be revealed in order to determine whether the graph has a given monotone property P or not (see also [5]).

3 Using hypercubes to approach the game

One of the interesting connections between hat guessing games and applications lies in interpreting the game in terms of various structure restrictions on hypercubes (the restriction depending on the rule of the particular game). In this section we will consider several questions for which we can use hypercubes to give an answer. One drawback to using hypercubes is that the number of vertices is exponential in the number of players and so many of the constructions are not polynomial in n. Nevertheless insight to the game can be achieved by considering hypercubes (see for example Proposition 13).

Recall that the n-cube has as vertices all 2^n binary words of length n. This has a natural connection with the possible placements of hats. The edges of the n-cube join two vertices which differ in one letter. As an example, in the 4-cube there is an edge between 1011 and 1010; we can represent this in shorthand as 101$*$ where the $*$ indicates an indeterminate bit which is either 0 or 1. The edges of the n-cube represent the decisions which must be made in forming the strategy. So 101$*$ indicates that the fourth player (the $*$) sees 1, 0, and 1 on the first, second, and third players respectively. In this situation they must either guess 0 or 1. To indicate his/her guess we will "orient" the edge in the direction of the guess. So for instance if the player guesses 0 in this case then we will have 1011→1010.

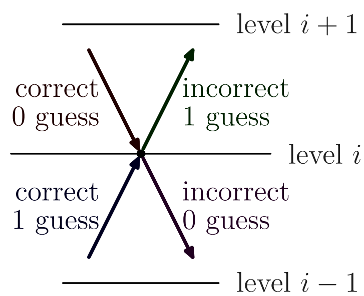
The original version of the game reduces to finding an orientation on the edges of the n-cube to maximize the minimum in-degree.

3.1 Balanced strategies

We now return to the original game. In Theorem 2 we gave one example of how to construct a strategy to guarantee $\lfloor n/k \rfloor$ guesses. This strategy is far from unique, and may not have some desired property. For example, while that strategy is easy to implement the correct guesses are not reflective of the actual hats that are placed on the players. For 2 colors of hats we will show how to construct different strategies and in particular how to construct a balanced strategy.

Lemma 11. *If there are n players and 2 different hat colors, then there exists a strategy which is balanced. That is, if there are b blue hats and r red hats placed on the players ($r + b = n$) then at least $\lfloor b/2 \rfloor$ of the people wearing blue guess correctly and $\lfloor r/2 \rfloor$ of the people wearing red guess correctly.*

We first approach the problem by using hypercubes. To construct a balanced strategy we will group the vertices of the hypercube in i levels, where a vertex is at level i if the word indexing the vertex has Hamming weight i. The up-degree (respectively down-degree) at a vertex in level i will be the number of edges between that vertex and vertices in level $i + 1$ (respectively $i - 1$). If we consider how directing the edges corresponds to guesses we have the following picture.



We now see that the balanced strategy in Lemma 11 would correspond to an orientation on the edges of the hypercube so that at each node the number of directed edges from level $i + 1$ to that node is $\lfloor$up-degree$/2\rfloor$ while the number of directed edges from level $i - 1$ to that node is $\lfloor$down-degree$/2\rfloor$.

Construction of Lemma 11. For n even we start with any edge and orient it arbitrarily and then continue to lengthen the directed path to be as long as possible by continually directing an undirected edge which is incident with the current terminal vertex. The only restriction is that if an edge is between level i and level $i+1$ then if possible the next edge will also be between level i and level $i + 1$. When the path can no longer be extended, if we have not oriented all the edges, then we pick an unoriented edge and repeat the process.

When n is odd a similar construction works, the only caveat being that we must be careful in selecting our initial edges. Now an initial edge cannot be chosen arbitrarily but must be directed up from a vertex of odd up-degree or down from a vertex of odd down-degree.

It is easy to see that the strategy we construct is balanced, since at each vertex we pair directed in- and out-edges, in such a way that every edge coming in from below (resp. above) is paired, if possible, with an edge going to below (resp. above). □

Using a different technique based on network flow, it is possible to construct balanced strategies for every k; see [1] for details.

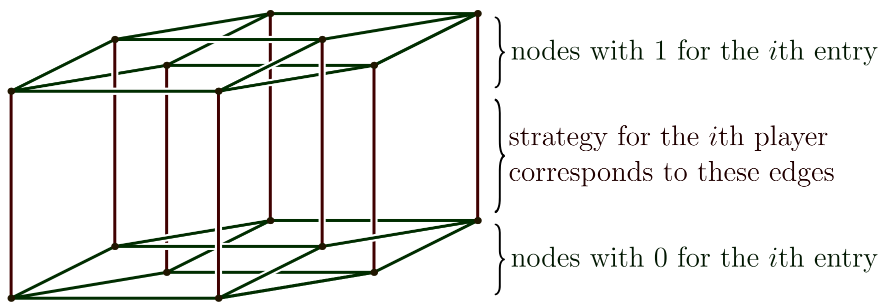
Theorem 12 ([1]). *If there are n players and k different hat colors, then there exists a strategy which is balanced. That is, if a_i of the players are wearing hats of color i ($1 \leq i \leq k$) then at least $\lfloor a_i/k \rfloor$ of the people wearing color i guess correctly for each value of i.*

3.2 Optimal strategies are unbiased

If one constructs many optimal strategies for the 2-color game when n is even, one starts to see a pattern emerge. Namely, each player is as likely to guess one hat color as they are to guess the other. We give a short proof that uses hypercubes.

Proposition 13. *Suppose the set of hat colors is $\{0, 1\}$ and there are n players playing an optimal strategy, where n is even. For any fixed player i, looking over all possible hat placements, we have that the number of times that player i guesses 0 is the same as the number of times that player i guesses 1.*

Proof. When n is even an optimal strategy corresponds to an orientation on the edges of the n-cube with the in-degree equal to the out-degree at each vertex. In particular, the strategy corresponds to some Eulerian walk on the n-cube. We now redraw the n-cube as follows.



Then note that the number of edges in the center that point up is the number of times the ith player guesses 1 and the number of edges that point down is the number of times that the ith player guesses 0. Since we have an Eulerian walk the number of up-edges equals the number of down-edges. $\square$

Proposition 13 can be generalized to games with more than 2 colors.

Proposition 14. *Suppose that n players are playing an optimal strategy of the k-color game, where k is a divisor of n. If the players' hat colors are drawn independently from the uniform distribution on $\{1, 2, \ldots, k\}$ then for each player i and each hat color c,*

$$\Pr(i \text{ guesses } c) = 1/k.$$

Equivalently, in an optimal strategy each player guesses each hat color an equal number of times.

Proof. Let X denote the random variable which counts the number of correct answers provided by the players. Our assumption that the players use an optimal guessing strategy means that $X \geq n/k$ at every point of the sample space. Now let $\mathcal{E}(i,c)$ denote the event that player i is assigned hat color c. We have

$$\begin{aligned} \mathbb{E}[X \mid \mathcal{E}(i,c)] &= \sum_{j=1}^{n} \Pr(j \text{ guesses correctly} \mid \mathcal{E}(i,c)) && (1) \\ &= \frac{n-1}{k} + \Pr(i \text{ guesses } c \mid \mathcal{E}(i,c)) && (2) \\ &= \frac{n-1}{k} + \Pr(i \text{ guesses } c). && (3) \end{aligned}$$

Here (1) follows from linearity of expectation, and (2) follows from the fact that, conditional on $\mathcal{E}(i,c)$, every player except i has a hat color which is uniformly distributed and is independent of its own guess. Finally, (3) follows from the fact that player i's guess is independent of its own hat color.

Recalling now that $X \geq n/k$ at every point of the sample space, we see that

$$\mathbb{E}[X \mid \mathcal{E}(i,s)] \geq n/k,$$

and according to (3) this implies $\Pr(i \text{ guesses } c) \geq 1/k$. Since this inequality applies to every hat color c, it must be the case that $\Pr(i \text{ guesses } c) = 1/k$ for every color c. □

3.3 The limited hats game

We now consider another variation on the original hats game. The setup is as before, but now the adversary has a limited supply of hats to choose from. We will let $H(n; a_1, a_2, \ldots, a_k)$ denote the maximum number of correct guesses that we can guarantee when there are n players and a_1 hats of the first color, a_2 hats of the second color, and so on up through a_k hats of the kth color. We need $a_1 + a_2 + \cdots + a_k \geq n$ (to ensure that we have enough hats for the players) and without loss of generality we can assume that $0 < a_i \leq n$ for all i.

Example 15. Suppose that there are 3 players and the adversary has 2 blue hats and 2 red hats. The players can choose to ignore this information and use the same strategy as in Theorem 2 guaranteeing 1 correct guess. However if they modify their strategy then they can guarantee 2 correct guesses. If the players are a, b, c then such a strategy would be for a to guess the opposite of what b is wearing, b to guess the opposite of what c is wearing, and c to guess the opposite of what a is wearing. So $H(3; 2, 2) = 2$. (More generally, it was shown in [3] that $H(4k-1; 2k, 2k) = 3k-1$.) □

Theorem 16. *We have the following properties:*

(i) $H(n; \underbrace{n, n, \ldots, n}_{k \text{ times}}) = \lfloor n/k \rfloor$.

(ii) If $a_1 + a_2 + \cdots + a_k = n$ *then* $H(n; a_1, a_2, \ldots, a_k) = n$.

(iii) *If m is even or k is odd (or both), then* $H(mk-1;\underbrace{m,\ldots,m}_{k\ \text{times}}) = \dfrac{mk+m-2}{2}$.

(iv) $H(n;a_1,a_2,\ldots,a_k) = H(n;a_{\sigma(1)},a_{\sigma(2)},\ldots,a_{\sigma(k)})$ *for any permutation* σ.

(v) *If* $b_i \le a_i$ *for all* $i=1,2,\ldots,k$ *then* $H(n;a_1,a_2,\ldots,a_k) \le H(n;b_1,b_2,\ldots,b_k)$.

(vi) $H(n;a_1,a_2,\ldots,a_k) \le \left\lfloor \displaystyle\sum_{\substack{b_i\le a_i,\,1\le i\le k\\ b_1+\cdots+b_k=n-1}} n\binom{n-1}{b_1,\ldots,b_k} \Big/ \sum_{\substack{b_i\le a_i,\,1\le i\le k\\ b_1+\cdots+b_k=n}} \binom{n}{b_1,\ldots,b_k}\right\rfloor$.

Proof. Item (i) is Theorem 2. Item (ii) is obvious because the strategy is to have each player guess the hat they do not see. Item (iv) says we can permute the hat colors. Item (v) follows by noting that the optimal strategy for the $H(n;a_1,a_2,\ldots,a_k)$ game is also a (not necessarily optimal) strategy for the $H(n;b_1,b_2,\ldots,b_k)$ game.

To prove item (vi) we first give a hyper-hypercube interpretation of the game. The k^n vertices are words of length n from the alphabet $\{0,\ldots,k-1\}$ and correspond to the k^n possible placements of hats. The tuples (i.e., edges) represent the decisions which must be made in deciding a strategy. So for example if $n=5$ and $k=3$ then one tuple would be $210{*}1 = \{21001, 21011, 21021\}$ representing the situation when the fourth player (the $*$) sees 2, 1, 0, and 1 on the first, second, third, and fifth players respectively. A strategy corresponds to marking one vertex on each tuple, the marking indicating the guess that the player will make. Note each marking is one correct guess.

We now use an averaging argument similar to that given in Theorem 2.

$$\begin{aligned}
H(n;a_1,a_2,\ldots,a_k) &\le \left\lfloor \frac{\#\text{ of correct guesses}}{\#\text{ of possible hat placements}} \right\rfloor \\
&= \left\lfloor \frac{\#\text{ of tuples available for marking}}{\#\text{ of vertices to be marked}} \right\rfloor \\
&= \left\lfloor \sum_{\substack{b_i\le a_i,\,1\le i\le k\\ b_1+\cdots+b_k=n-1}} n\binom{n-1}{b_1,\ldots,b_k} \Big/ \sum_{\substack{b_i\le a_i,\,1\le i\le k\\ b_1+\cdots+b_k=n}} \binom{n}{b_1,\ldots,b_k} \right\rfloor
\end{aligned}$$

The numerator is the n possible positions of the $*$ along with the allowable combinations of the remaining $n-1$ entries. The denominator is the number of ways to place the n hats in allowable combinations.

For item (iii) we have that $H(mk-1;m,\ldots,m) \le (mk+m-2)/2$ from (vi). So it suffices to show that we can construct a strategy guaranteeing at least $(mk+m-2)/2$ correct guesses. There are two types of tuples, those which involve only one markable vertex and those that involve two. The first kind is for players who see a full set of all but one type of hat and so automatically know their hat. The second kind is for players who see a full set of all but two types of hats and so have to make one of two choices.

Every vertex will be associated with $m-1$ tuples of the first type (one for each hat of the deficient color in the placement), we mark these tuples and put them aside. We now construct a bipartite graph with the remaining edges and tuples as follows: The vertex set is $S \cup T$ where S

is the set of tuples we have not yet marked and T is the set of vertices to be marked, there is an edge between an element in S and an element in T if the corresponding vertex can be marked by the corresponding tuple.

Every element in S has degree 2 and every element in T has degree $mk-m$, which by assumption is even. We now split the elements in T by duplicating each element $(mk-m)/2$ times and then distributing the edges of the original element so that each resulting piece has degree 2. It is easy to now construct a perfect matching between S and T (for example start with any edge and going through a cycle alternatively include/not include the edges). This perfect matching gives a marking on the tuples so that each of the vertices is marked $(mk-m)/2$ times.

In total each vertex was marked $(m-1)+(mk-m)/2 = (mk+m-2)/2$ times giving our desired strategy. □

Question. What is $H(n; a_1, a_2, \dots, a_k)$? Is the upper bound given in Theorem 16 tight? [Note: items (i), (ii) and (iii) in Theorem 16 are examples where the bound is tight.]

As a warmup to the above question, the interested reader might enjoy showing that $H(5; 4, 3) = 3$ (an upper bound of 3 immediately follows from Theorem 16 so it suffices to find a strategy guaranteeing at least 3 correct guesses).

4 Summary and open questions

In this paper we considered hat guessing games in which players wearing hats of various colors use deterministic strategies to guess the color of some hats (usually their own) with the goal of maximizing the number of correct answers in the worst case. In this section we summarize our main results, and we list the open questions which are scattered throughout this paper.

Focusing on hat games with a *sight graph* that specifies the set of hats visible to each player, we defined the *hat number* $H_k(G)$ to be the number of correct answers provided in the worst case by an optimal strategy for players guessing their own hat color, when the sight graph is G and there are k colors of hats. We proved that for two colors and undirected sight graphs, the hat number equals the cardinality of a maximum matching in the sight graph. For a directed graph G and two hat colors, we provided lower and upper bounds on the hat number; these bounds suffice to distinguish graph families with bounded hat number from those with unbounded hat number. For three or more colors, we proved that the hat number of a tree is zero and that there are bipartite graphs with nonzero hat number. When there are two hat colors and there is a *guessing graph* which specifies the set of hat colors which each player must guess, we provided necessary and sufficient conditions for the existence of a strategy which guarantees at least one correct answer.

Turning to questions about the distribution of guesses and of correct guesses, we introduced a hypercube interpretation of the game which permitted us to prove that when there are two colors of hats and players can see all hats except their own, there is a guessing strategy which guarantees that roughly half of the players wearing each type of hat guess correctly. (This result was also proved in [1], which contains a generalization to more than two colors.) We also proved that optimal strategies are unbiased, i.e. when the number of hat colors is a divisor of the number of players, each player guesses each hat color with equal probability when the colors are assigned independently and uniformly at random. Finally, turning to a version in which the players are given an *a priori* upper bound on the number of hats of each color, we exhibited some bounds on the number of correct answers provided by the optimal strategy in the worst case.

In presenting these results, we also introduced many open questions inspired by them. Here we recapitulate the open questions presented earlier.

Complexity of computing hat numbers and optimal strategies: What is the computational complexity of determining $H_k(G)$, given k and G? There is a polynomial-time algorithm when $k = 2$ and G is undirected (because a maximum matching can be computed in polynomial time), but for all other cases the best known algorithm requires nondeterministic exponential time. On the other hand, we do not know if the problem is NP-hard.

How much computational power is required to implement an optimal guessing strategy? For $k = 2$, can the optimal guessing strategy for a directed sight graph G always be implemented by players using Boolean circuits of size polynomial in the size of G?

Graphs with positive hat numbers: Is there a bipartite graph G satisfying $H_k(G) > 0$ whose size is polynomial in k? What if instead of bipartite we consider k-clique-free graphs? Are there infinitely many graphs G such that $H_k(G) > 0$ but $H_k(G') = 0$ for every proper subgraph $G' \subset G$?

Augmenting sight graphs to improve hat numbers: Given a positive number m and two graphs G_v, G_g (the sight graph and guessing graph of a hat guessing game), determine the smallest value j such that one can add j additional edges to G_v to obtain a graph G'_v satisfying $H_2(G'_v, G_g) \geq m$.

The limited hats game with unrestricted vision: Suppose players can see every hat color except their own, and must guess their own hat color. When there are k hat colors and at most a_i hats in color class i ($1 \leq i \leq k$), compute the maximum number of correct answers that can be guaranteed by a guessing strategy given this limitation on the placement of hats. Is the upper bound in part (vi) of Theorem 16 tight?

5 Acknowledgements

We thank Joe Buhler, Kevin Costello, Harvey Friedman, and Ron Graham for interesting and helpful discussions on this subject.

References

[1] G. Agarwal, A. Fiat, A. V. Goldberg, J. D. Hartline, N. Immorlica, and M. Sudan. Derandomization of auctions. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC)*, pages 619–625, 2005.

[2] C. Berge. Sur le couplage maximum d'un graphe. *Comptes Rendues Hebdomadaires des Séances de l'Académie des Sciences [Paris]*, 247:258–259, 1958.

[3] U. Feige. You can leave your hat on (if you guess its color). *Technical Report MCS04-03, Computer Science and Applied Mathematics, The Weizmann Institute of Science*, page 10 pp., 2004.

[4] B. Reed, N. Robertson, P. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.

[5] R. L. Rivest and J. Vuillemin. A generalization and proof of the aanderaa-rosenberg conjecture. In *STOC '75: Proceedings of seventh annual ACM symposium on Theory of computing*, pages 6–11, New York, NY, USA, 1975. ACM Press.

[6] S. Robinson. Why mathematicians now care about their hat color. *The New York Times*, page D5, April 10, 2001.

[7] A. L. Rosenberg. On the time required to recognize properties of graphs: A problem. *SIGACT News*, 5, 1973.

[8] W. T. Tutte. The factorization of linear graphs. *J. London Math. Society*, 22:107–111, 1947.

[9] P. Winkler. Games people don't play. In *Puzzlers' Tribute*. A. K. Peters, 2001.