Cooperation via Codes in Restricted Hat Guessing Games

Kai Jin*
HKUST
Hong Kong, China
cscjjk@gmail.com

Ce Jin
Tsinghua University
Beijing, China
jinc16@mails.tsinghua.edu.cn

Zhaoquan Gu†
Guangzhou University
Guangzhou, China
zqgu@gzhu.edu.cn

ABSTRACT

Hat guessing games have drawn a lot of attention among mathematicians, computer scientists, coding theorists and even the mass press, due to their relations to graph theory, circuit complexity, network coding, and auctions. In this paper, we investigate a new variant where there is exactly one hat of each color and where each player may receive multiple hats. Assume there are n players and T hats with different colors. A dealer randomly places k hats to each player and holds $T-nk$ hats in hand. After observing the (colors of) hats of other players but not those of themselves, the players shall guess their colors simultaneously by a pre-coordinated strategy. We present methods to compute the best strategy under two common winning rules: *all guesses are right* or *at least one guess is right*, and derive exact value of the maximum winning probability for several cases. Especially, we introduce a novel notion called Latin matching between $\binom{[2n-1]}{n-1}$ and $\binom{[2n-1]}{n}$ and establish its connection to the solution of some restricted cases. Here, $\binom{[2n-1]}{n-1}$ (respectively, $\binom{[2n-1]}{n}$) denotes the set of $(n-1)$-element (respectively, n-element) subsets of $\{1,\ldots,2n-1\}$. Moreover, we show that some well-known combinatorial results (e.g. the antipodal matching between two symmetric layers of the subset lattice and the ordered design $OD(t,k,v)$ given in modern design theory) can be applied to design explicit strategies in other cases. From our results we observe an interesting phenomenon that a leader is necessary for consensus but unnecessary for decentralization.

KEYWORDS

Agent cooperation; Hat guessing game; Theory of error-correcting code; Ordered design; Hamming code

ACM Reference Format:
Kai Jin, Ce Jin, and Zhaoquan Gu. 2019. Cooperation via Codes in Restricted Hat Guessing Games. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Hat guessing games have been studied extensively in recent years due to their close relations to graph entropy, circuit complexity, network coding, and auctions [6, 7, 12–14, 20, 21, 26]. In particular, some versions of hat guessing games are used as toy problems in de-randomizing protocols in circuit complexity and de-randomizing auctions in auction mechanisms design, due to the innate similarities between these games and the number-on-forehead models in complexity [3, 11] and the bid-independent auctions [1, 4]. Other researchers are attracted by these games because the optimal solutions of several variants have unexpected connections to coding theory. For example, the solutions in [3, 11] can be constructed from the well-known Hamming codes.

In this paper, we study a unique-supply hat guessing game (defined below), which is a special case of the finite-supply variant proposed and studied in [7, 12], in which the number of hats are limited (which is more realistic). Unlike the original version, best strategies based on combinatorial codes for this variant have not been reported in prior work. Therefore, investigating such strategies seems important at the moment and will be the main agenda of this paper.

- Assume there are *n players* and *T hats* with different colors $1,\ldots,T$. (There is exactly one hat of each color; in other words, the supply for hats of any specific color is unique.) A *dealer* places k hats to each of the n players according to the uniform distribution, and $d = T - nk \geq 1$ hats remain in the dealer's hand.
- Each player knows the colors of hats of other players but cannot see and has to guess all the k colors he or she receives. The answer of any player consists of exactly k different colors. The answer is *right* if all the k colors are correct. All players guess simultaneously.
- The n players act as a *team*. The team wins or loses, not individuals. There are two different winning rules:
  – All-right: The team wins if all players are right.
  – One-right: The team wins if at least one is right.
- The parameters (n,k,d) are known to the players (and $T = d + nk$). No communication is allowed between team members after the game starts. Communication via wait as used in some hat puzzles [6] is also forbidden because all players guess simultaneously. However, it is permissible for them to discuss a strategy beforehand.
- The question is how to design the best cooperative strategy to get the maximum winning probability.

Example 1.1. Assume $(n,k,d) = (2,1,2)$. If player 1 guesses color (of her own hat) as $(B-1) \mod 4$ after observing

[*] Corresponding author.
[†] Supported by grant U1636215 and National Key R&D Program of China 2018YEB1004003. Corresponding author.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

the hat color B of its teammate, and player 2 guesses color (of his own hat) as $(A+1) \mod 4$ after observing the hat color A of its teammate, this joint strategy is the best for the all-right winning rule, as these two players either guess right simultaneously or both make wrong guesses.

Our results. 1. We present general methods to compute the best strategy for both winning rules, and prove exact values of the maximum winning probability for all cases under the one-right rule and some cases under the all-right rule.

2. We show that constructing explicit best strategies via codes leads to some subtle combinatorial problems. Especially, in one of our constructions (given in subsection 2.1), we introduce a notion called *Latin* matching between $\binom{[2n-1]}{n-1}$ and $\binom{[2n-1]}{n}$, which is worth to study in its own right. Here, $\binom{[2n-1]}{n-1}$ (respectively, $\binom{[2n-1]}{n}$) denotes the set of $(n-1)$-element (respectively, n-element) subsets of $\{1,\ldots,2n-1\}$. (Yet our best strategy via codes does not always exist due to the lack of the code for particular parameters n,k,d.)

3. We discuss the application of our best strategies in derandomization (in Section 4). Also, by comparing our best strategies under two different rules, we discuss the necessity of having a leader for making consensus in team cooperation.

A comparison of the models. A player in our game can infer some information about his own color, e.g., c is not his color if he observes that color c is placed to another player, whereas in the unlimited-supply version a player knows nothing about his color. Another major difference is that we allow $k \geq 1$ whereas the previous models only consider the case $k=1$.

1.1 Preliminary

Let $v_1,\ldots,v_s$ denote all the s possible placements of hats, where $s = \frac{T!}{d!(k!)^n}$. For each player $i \in [n]$, the number of his or her possible observations is $m = s/\binom{k+d}{d}$. Denote his or her possible observations by $u_{i,1},\ldots,u_{i,m}$.

Define a bipartite graph $H(n,k,d) = (U,V,E)$ as follows: $U = \bigcup_{i=1}^{n}\{u_{i,1},\ldots,u_{i,m}\}$, $V = \{v_1,\ldots,v_s\}$, and $(u_{i,j},v_k) \in E$ if and only if placement v_k leads to observation $u_{i,j}$. Clearly, each vertex in U has degree $\binom{k+d}{d}$, and each vertex in V has degree n. More precisely, given any vertex v_k, there exists exactly one j for each $i \in [n]$ so that v_k is linked to $u_{i,j}$. For example, see $H(2,1,2)$ in the dashed box in Figure 1.

Denote $V_G = \{v_1,\ldots,v_s\}$ and build a graph $G(n,k,d) = (V_G,E_G)$ as follows. $(v_k,v_l) \in E_G$ if the distance between v_k and v_l is **two** in $H(n,k,d)$; equivalently, $(v_k,v_l) \in E_G$ if there exists a vertex $u_{i,j}$ such that $(u_{i,j},v_k) \in E$ and $(u_{i,j},v_l) \in E$. Intuitively, $(v_k,v_l) \in E_G$ means that there exists a player who cannot distinguish the two placements v_k and v_l.

The *subset lattice* is the family of all subsets of $[T]$, partially ordered by inclusion. The k-th layer of this lattice, denoted by $\mathcal{P}_k$, consists of all of the k-element subsets of $[T]$.

2 ALL-RIGHT WINNING RULE

Let $p'_{\max}(n,k,d)$ (or $p'_{\max}$ when (n,k,d) are clear) be the maximum winning probability for the all-right winning rule.

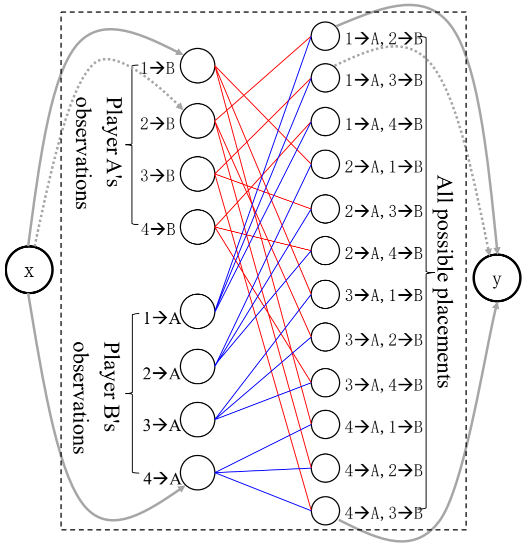

Figure 1: $H(2,1,2)$. Players are denoted by A and B.

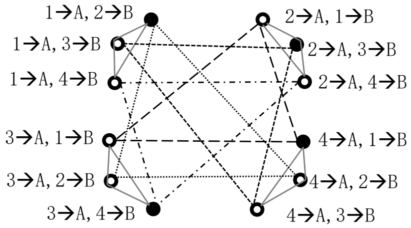

Figure 2: $G(2,1,2)$. Players are denoted by A and B.

Proposition 2.1. $p'_{\max} \leq 1/\binom{k+d}{d}$.

Proof. Consider an arbitrary strategy S. The probability that player 1 (we may consider any other player instead) guesses right is $1/\binom{k+d}{d}$. This means that the probability that all players are right is at most $1/\binom{k+d}{d}$ using S. □

The next lemma gives a graph interpretation of the game.

Lemma 2.2. *Let $\alpha(G)$ be the cardinality of the maximum independent (vertex) set of G. Then, $p'_{\max} = \alpha(G(n,k,d))/s$. Moreover, the best strategy can be easily computed provided that a maximum independent set of $G(n,k,d)$ is given.*

Proof. Given an independent set I of $G(n,k,d)$ (note that each vertex in I is a possible placement), we can design a strategy that wins all placements in I as follows. For each player, his or her observation (which is a vertex $u_{i,j}$ in $H(n,k,d)$) is linked to at most one node in I. He or she answers this node, if any; and answers arbitrarily otherwise. On the contrary, given any strategy, from its winning placements we immediately obtain an independent set of $G(n,k,d)$. □

Example 2.3. $(n,k,d) = (2,1,2)$. We have $\alpha(G(2,1,2)) = 4$: a maximum independent set is drawn in Figure 2 (indicated by the four solid circles). So, $p'_{\max} = 4/12$ by Lemma 2.2.

By Lemma 2.2, computing $p'_{\max}$ and the best strategy under all-right winning rule reduces to computing the maximum independent set of $G(n, k, d)$. Unfortunately, we have no idea how to compute this set in polynomial time of s, or only derive a formula for quantity $\alpha(G(n, k, d))$, even though $G(n, k, d)$ is highly symmetric (a vertex-transitive graph indeed).

One may guess that $p'_{\max} = 1/\binom{k+d}{d}$. But this is not true even for $k = 1$. By computer programs [19] we verified that $G(4, 1, 3)$ does not admit an independent set with size $s/4$, which means $p'_{\max} < 1/4 = 1/\binom{k+d}{d}$ for $(n, k, d) = (4, 1, 3)$.

An outline of this section. Subsection 2.1 introduces a novel notion called Latin matching and presents Latin matchings of order 2, 3, and 5. Briefly, a Latin matching of order n is a perfect matching from $\binom{[2n-1]}{n-1}$ to $\binom{[2n-1]}{n}$ with an additional property analogous to the property of Latin square. Subsection 2.2 constructs a maximum independent set of $G(n, 1, n-1)$ – i.e, it constructs an optimal strategy for the case $(n, k, d) = (n, 1, n-1)$ under all-right rule – based on a Latin matching of order n. Subsection 2.3 proves two results concerning the Latin matchings of higher orders. One states that constructing cyclic Latin (see definition below) matching is as hard as constructing the Steiner system $S(n-2, n-1, 2n-2)$. The other states that the order of a Latin matching must be prime. Subsection 2.4 discusses why we focus on the case $(n, 1, n-1)$ and discusses other combinations of (n, k, d). Particularly, we use an ordered design [8, 10, 23] (which is a kind of combinatorial design) to solve some cases of $k = 1$.

2.1 Introduction of Latin Matchings

Definition 2.4. Assume f is a perfect matching from $\binom{[2n-1]}{n-1}$ to $\binom{[2n-1]}{n}$ in the subset lattice (set A can be mapped to A' only if $A \subset A'$). We say f is of ***order*** n. Let $f^+(\{i_1, \ldots, i_{n-1}\})$ denote the only element in $f(\{i_1, \ldots, i_{n-1}\}) - \{i_1, \ldots, i_{n-1}\}$. We say f is ***Latin*** if $f^+(\{i_1, \ldots, i_{n-1}\}) \neq f^+(\{j_1, \ldots, j_{n-1}\})$ whenever $\{i_1, \ldots, i_{n-1}\}$ and $\{j_1, \ldots, j_{n-1}\}$ share exactly $n-2$ common elements. Moreover, we say f is ***cyclic*** if

$$f^+(\{i_1 + 1, \ldots, i_{n-1} + 1\}) \equiv f^+(\{i_1, \ldots, i_{n-1}\}) + 1,$$

where numbers are taken modulo by $(2n - 1)$.

Example 2.5. Here is a Latin matching f of order 2: $f(\{1\}) = \{1, 2\}, f(\{2\}) = \{2, 3\}, f(\{3\}) = \{3, 1\}$. A Latin matching f' of order 3 is shown in Figure 3. The left table shows $(f')^+$, whose Latin property is easy to check.

$(f')^+$	1	2	3	4	5
1	–	4	2	5	3
2	4	–	5	3	1
3	2	5	–	1	4
4	5	3	1	–	2
5	3	1	4	2	–

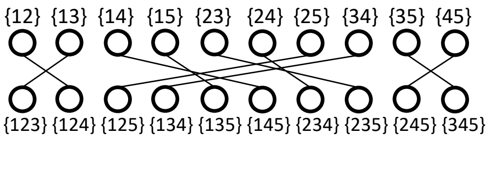

Figure 3: A Latin matching of order 3.

Example 2.6. We demonstrate a cyclic Latin matching f of order 5 by Figure 4. Since cyclic, it can be illustrated by $\binom{9}{4}/9 = 14$ pictures. In each picture, we label the small circles by 1 to 9 in clockwise order, ending at the topmost one. Assuming the four solid circles are labeled by i_1, i_2, i_3, i_4 respectively, then this picture indicates that $f^+(\{i_1, i_2, i_3, i_4\}) = 9$. For example, the first four pictures in the first row indicate that f^+ maps $\{3, 4, 5, 6\}$, $\{2, 4, 5, 7\}$, $\{1, 4, 5, 8\}$, and $\{2, 3, 4, 8\}$ to 9. The next lemma proves that f is indeed a Latin perfect matching and reveals the connection between f and the Hamming code $\mathsf{Hamming}(8, 4, 4)$.

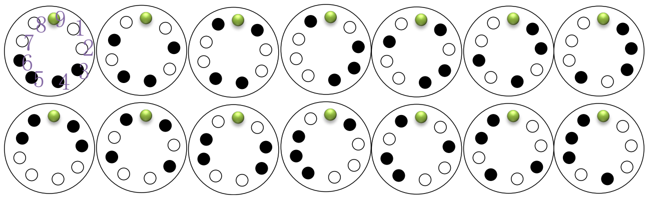

Figure 4: A Latin matching of order 5.

Lemma 2.7. *1. The matching f defined above is a perfect matching from $\binom{[9]}{4}$ to $\binom{[9]}{5}$. 2. This matching is Latin.*

Proof. In the 14 pictures, the shapes of the four black circles are different under rotating, so as the shapes of the four white circles. This implies the first part of this lemma.

To prove that f is Latin, we first point out a connection between f and $\mathsf{Hamming}(8, 4, 4)$. Notice that we can represent each of the 14 pictures by a binary word $a_1 a_2 \ldots a_8$ of length 8 (e.g., 00111100 and 01011010 for the first two pictures), and we claim that these 14 words are exactly those 14 codewords of $\mathsf{Hamming}(8, 4, 4)$ which are not all 0's and not all 1's. [1] Precisely, they can be described by the following equations.

$$\begin{cases} a_5 = & a_1 \oplus a_2 \oplus a_3, \quad a_6 = & a_1 \oplus a_2 \oplus a_4, \\ a_7 = & a_1 \oplus a_3 \oplus a_4, \quad a_8 = & a_2 \oplus a_3 \oplus a_4. \end{cases} \tag{1}$$

A well-known property of $\mathsf{Hamming}(8, 4, 4)$ is that the hamming distance between the codewords is at least 4. This property is useful in proving the Latin property of f:

Suppose to the opposite that f is not Latin. There should be two 4-element subsets A, B of $[9]$ with exactly three common elements such that they are both mapped to 9 under f^+. If we translate A and B to the length 8 binary codes a and b, we know that $\mathsf{d}(a, b) \geq 4$ according to the above correspondence between f and $\mathsf{Hamming}(8, 4, 4)$, where $\mathsf{d}(a, b)$ denotes the hamming distance between a and b. On the other hand, $\mathsf{d}(a, b) = 2$ since sets A, B are of size 4 and they share exactly three common elements. Contradictory. □

As a summing up of this subsection, we have shown

Lemma 2.8. *There exist Latin matchings of orders* 2, 3, 5.

Two other cyclic Latin matchings of order 5 are drawn in Figure 5. Interestingly, these two matchings can also be represented by $\mathsf{Hamming}(8, 4, 4)$ (proof omitted).

[1] The original Hamming code is slightly different. But it is equivalent to the one given here under permutation. The one in (1) admits an interesting symmetric property which is not admitted by the original one: if $a_1 \ldots a_8$ is a codeword, its reverse $a_8 \ldots a_1$ is also a codeword.

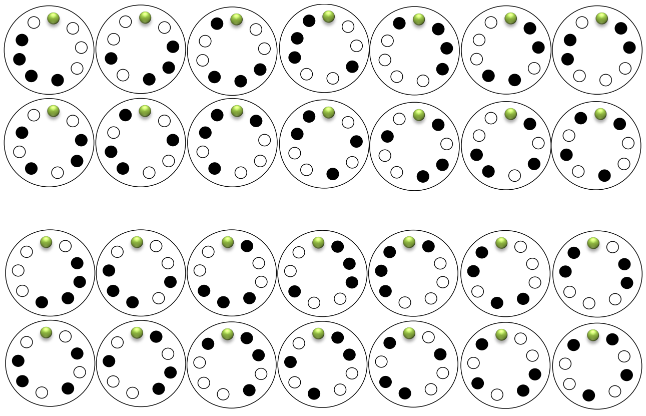

Figure 5: Two more Latin matchings of order 5**.**

2.2 Optimal Strategy via Latin Matching

Lemma 2.9. *The existence of a Latin matching of order* n *implies the following: (1)* $G(n,1,n-1)$ *is* n*-colorable; (2)* $\alpha(G(n,1,n-1)) = s/n$*; and (3)* $p'_{\max}(n,1,n-1) = 1/n$*. Moreover, we can design an optimal strategy for the case of* $(n,k,d) = (n,1,n-1)$ *using a Latin matching of order* n*.*

Before proving this lemma, we reveal a connection between the n-colorability of $G(n,1,n-1)$ and the Latin square problem. Recall that each vertex of $G(n,1,n-1)$ corresponds to a placement of the T hats (notice that $T = 2n-1$ when $k = 1$ and $d = n-1$) to n players, each of which receives one hat. Such a placement can be represented as an n-dimensional vector $\mathbf{a} = (a_1,\ldots,a_n)$ where $a_1,\ldots,a_n \in [2n-1]$ and $a_i \neq a_j$ $(\forall i \neq j)$, and where a_i indicates the hat that is given to player i. Two vertices $\mathbf{a} = (a_1,\ldots,a_n)$ and $\mathbf{b} = (b_1,\ldots,b_n)$ are connected in E_G if vectors $\mathbf{a},\mathbf{b}$ have *hamming distance* 1, i.e. $\sum_{i=1}^{n}(1-\delta_{a_ib_i}) = 1$, where δ is the *Kronecker delta*.

Moreover, consider an incomplete n-dimensional hypercube as follows. Initially, this hypercube has side length $2n-1$ and consists of $(2n-1)^n$ unit cells, each of which is denoted by a coordinate $(a_1,\ldots,a_n)$ where $a_1,\ldots,a_n \in [2n-1]$. Then, remove the cells for which $a_1,\ldots,a_n$ are not distinct. As a result, the vertices in $G(n,1,n-1)$ correspond to the remaining cells, and two vertices are connected if and only if their corresponding cells are in the same *orthogonal line* (a line that is parallel to one of the n axes is said orthogonal).

According to the above analysis, finding a vertex-color of $G(n,1,n-1)$ is equivalent to coloring the cells of the (incomplete) hypercube such that any cells in the same orthogonal line have different color. Moreover, if it is constrained to use n colors, the colors of cells in any orthogonal line must be a permutation of all n colors. Therefore, this is a variant of the celebrated Latin square problem in high dimension [9, 22].

Proof of Lemma 2.9. (1) Let f be a Latin matching of order n. We color the vertices of $G(n,1,n-1)$ as follows. Take any vertex $\mathbf{a} = (a_1,\ldots,a_n)$ where $a_1,\ldots,a_n$ follow the assumption in the analysis above. Let $S_{\mathbf{a}} = \{a_1,\ldots,a_n\}$. Because f is a perfect matching from $\binom{[2n-1]}{n-1}$ to $\binom{[2n-1]}{n}$, there must be a unique $i \in [n]$ such that $f(S_{\mathbf{a}} - \{a_i\}) = S_{\mathbf{a}}$. Define the color of $\mathbf{a}$ (denoted by $c(\mathsf{a})$) as i.

Assume now $\mathbf{a}$ and $\mathbf{b}$ are two adjacent vertices. So, there is a unique $j \in [n]$ such that $a_j \neq b_j$. We shall prove that $c(\mathsf{a}) \neq c(\mathsf{b})$. Suppose to the opposite that $c(\mathsf{a}) = c(\mathsf{b}) = i$.

Because $\mathbf{a}$ and $\mathbf{b}$ are adjacent, $S_{\mathbf{a}} \neq S_{\mathbf{b}}$. Because $c(\mathsf{a}) = i$, set $S_{\mathbf{a}} - \{a_i\}$ is mapped to $S_{\mathbf{a}}$ under f. Because $c(\mathsf{b}) = i$, set $S_{\mathbf{b}} - \{b_i\}$ is mapped to $S_{\mathbf{b}}$ under f. Altogether and since f is a bijection, $S_{\mathbf{a}} - \{a_i\} \neq S_{\mathbf{b}} - \{b_i\}$. This further implies that $j \neq i$ since $S_{\mathbf{a}} - \{a_j\} = S_{\mathbf{b}} - \{b_j\}$ by assumption of j.

Because $j \neq i$ and j is the unique subscript such that $a_j \neq b_j$, we know (i) $a_i = b_i$ and (ii) $S_{\mathbf{a}} - \{a_i\}$ shares exactly $n-2$ common elements with $S_{\mathbf{b}} - \{b_i\}$. The latter means $f^+(S_{\mathbf{a}} - \{a_i\}) \neq f^+(S_{\mathbf{b}} - \{b_i\})$ by the Latin property of f. Recall that $f(S_{\mathbf{a}} - \{a_i\}) = S_{\mathbf{a}}$. This means $f^+(S_{\mathbf{a}} - \{a_i\}) = a_i$. Recall that $f(S_{\mathbf{b}} - \{b_i\}) = S_{\mathbf{b}}$. This means $f^+(S_{\mathbf{b}} - \{b_i\}) = b_i$. Together, (iii) $a_i \neq b_i$, which is contradictory with (i).

(2) By Proposition 2.1, $p'_{\max} \leq 1/n$. Combining this with Lemma 2.2, $\alpha(G(n,1,n-1)) \leq s/n$. On the other side, since $G(n,1,n-1)$ is n-colorable, $\alpha(G(n,1,n-1)) \geq s/n$.

(3) This follows from (2) and Lemma 2.2.

In the following, we describe n optimal strategies. For each $i \in [n]$, let I_i denote the set of vertices with color i. Formally,

$$\begin{aligned}
I_1 &= \left\{(x, i_1, i_2, \ldots, i_{n-1}) \mid \{i_1,\ldots,i_{n-1}\} \in \tbinom{[2n-1]}{n-1}\right\},\\
I_2 &= \left\{(i_1, x, i_2, \ldots, i_{n-1}) \mid \{i_1,\ldots,i_{n-1}\} \in \tbinom{[2n-1]}{n-1}\right\},\\
&\cdots\\
I_n &= \left\{(i_1, i_2, \ldots, i_{n-1}, x) \mid \{i_1,\ldots,i_{n-1}\} \in \tbinom{[2n-1]}{n-1}\right\},
\end{aligned}$$

where x denotes $f^+(\{i_1,\ldots,i_{n-1}\})$.

For each $c \in [n]$, set I_c is a maximum independent set of $G(n,1,n-1)$. Applying Lemma 2.2, this means there is an optimal strategy so that the nodes in I_c are winning placements. We describe this strategy below more explicitly.

The explicit strategy corresponding to I_c. Assume player i receives the hat with color a_i. Write $\mathbf{a} = (a_1,\ldots,a_n)$. For each $i \in [n]$, denote $A_i = S_{\mathbf{a}} - \{a_i\}$, which is the set of hats observed by player i. In this strategy, we distinguish player c from the other players. For player c, she guesses $f^+(A_c)$. For player i where $i \neq c$, she plays as follows. Notice that a_c is known by player i. Since $A_i - \{a_c\}$ has size $n-2$ and due to Latin property, there exists a unique number $b_i \notin (A_i - \{a_c\})$ such that $f^+((A_i - \{a_c\}) + \{b_i\}) = a_c$. Player i guesses b_i.

We shall show that every player guesses right when $\mathbf{a} \in I_c$. This reduces to showing that (i) $f^+(A_c) = a_c$ and (ii) $b_i = a_i$ when $\mathbf{a} \in I_c$. Assume $\mathbf{a} \in I_c$. Fact (i) holds by the definition of I_c. Since b_i is the unique number $\notin (A_i - \{a_c\})$ such that $f^+((A_i - \{a_c\}) + \{b_i\}) = a_c$, whereas $a_i \notin (A_i - \{a_c\})$ and $f^+((A_i - \{a_c\}) + \{a_i\}) = f^+(A_c) = a_c$, we get (ii). □

Leadership in the above strategies. In this strategy, player c can be regarded as the **leader** of the team. This does not mean that she or he will answer earlier than the others. Instead, everybody considers what the leader would do and searches the unique answer consistent with the leader.

Remark. *1. Latin squares have wide applications in combinatorics, statistics, and computer science [9, 16]. They can be used for scheduling tournaments and processors of massively parallel computer. The n-coloring of $G(n,1,n-1)$ shown above may have analogous and even more interesting applications, since it is essentially a Latin hypercube.*

2. Let $\mathbf{a} = (a_1, \dots, a_n)$ denote a vertex in $G(n,1,n-1)$. The color of $\mathbf{a}$ is $c(\mathbf{a})$. Because $c(\mathbf{a})$ changes whenever exactly one of $a_1, \dots, a_n$ changes, this value can serve as the one-error detecting bit when we send message $a_1, \dots, a_n$.

3. The application of the Latin matching in our optimal strategy has the same favor as the application of the Hamming code in the optimal strategy of the original hat guessing game. The independent set defined from a Latin matching (e.g. I_1 or I_n) has the same favor as the hamming code in some sense.

2.3 Latin Matchings of Higher Orders

A *Steiner system* with parameters t, k, n, written as $S(t,k,n)$, is an n-element set S together with a set of k-element subsets of S, called blocks, with the property that each t-element subset of S is contained in exactly one block [5, 8, 24].

Theorem 2.10. *From a cyclic Latin matching of order n, we can obtain a Steiner system $S(n-2, n-1, 2n-2)$. In other words, finding a cyclic Latin matching of order n is at least as difficult as designing $S(n-2, n-1, 2n-2)$.*

Theorem 2.11. *The order of a Latin matching is prime.*

Remark. *1. The existence of $S(n-2, n-1, 2n-2)$ for $n > 11$ has been open for decades and is surprisingly difficult [5, 24]. A trivial prerequisite is that n must be prime. So the order of a cyclic Latin matching is prime. But this is covered by Theorem 2.11. Moreover, system $S(9,10,20)$ does not exist [24]. So, by Theorem 2.10, **there is no cyclic Latin matching of order 11**. Furthermore, system $S(5,6,12)$ does exist and is unique up to isomorphism and is widely known as the small Witt design W_{12} [24]. Yet **there is no cyclic Latin matching of order** 7 (see the discussion below).*

2. Every known Latin matching is isomorphic to a cyclic Latin matching. Whether this is always true is not known.

Proof of Theorem 2.10. According to our construction of the cyclic Latin matching of order 5 in subsection 2.1, a cyclic Latin matching f of order n can be represented by $M = \binom{2n-1}{n-1}/(2n-1)$ codewords of length $2n-2$ so that

1st. *Each codeword consists of half zeros and half ones.*

2nd. *If a and b are two codewords, then $a +' 0'$ is not cyclic equivalent to $b+'0'$ and $a+'1'$ is not cyclic equivalent to $b +' 1'$, where $+$ indicates the concatenation of strings.* (Note: This property implies that f is perfect.)

3rd. *The hamming distance between two codewords is at least 4.* (Note: This property implies that f is Latin.)

Let $S = \{1, \dots, 2n-2\}$. We build M blocks ($(n-1)$-element sets) of S from these codewords in a standard way. For each codeword $a = a_1 \dots a_{2n-2}$, we build a block $\{i \mid a_i = 1\}$. Since a has half ones, this block has $n-1$ elements indeed.

We now verify that each $(n-2)$-element set of S is contained in exactly one block. First, we claim that any $(n-2)$-element set is contained in at most one block. This is because when two blocks both contain the same $n-2$ elements, their corresponding codewords have hamming distance 2. Moreover, since each block has $n-1$ elements, the number of $(n-2)$-element sets covered by each block is $n-1$. So, at least $M \times (n-1) = \binom{2n-2}{n-2}$ different $(n-2)$-element sets are covered. This means every $(n-2)$-element set is covered.

The above process can be reversed. From a Steiner system $S(n-2, n-1, 2n-2)$, we can obtain M codewords satisfying the 1st and 3rd properties, but not necessarily the 2nd. □

Why is there no cyclic Latin matching of order 7? Finding such a matching reduces to finding a code satisfying the three properties above. By computer programs [18] we generate all codes admitting the 1st and 3rd properties (from the unique $S(5,6,12)$ using the reverse process mentioned above) and test if any of them admits the 2nd. The result is negative.

The remaining part of this subsection proves Theorem 2.11.

Proposition 2.12. *Given integer $n \geq 2$. It is prime if and only if $k!$ divides $(n+(k-1)) \times \ldots \times (n+1)$ for $2 \leq k \leq n-1$.*

Proof. Assume n is composite. Choose its prime factor k. Obviously, k does not divide $(n+(k-1)) \times \ldots \times (n+1)$. Thus $k!$ does not divide it either.

Assume n is prime. For $2 \leq k \leq n-1$, because $\binom{n+k-1}{k} = (n+(k-1)) \times \ldots \times n/k!$, we know $k!$ divides $(n+(k-1)) \times \ldots \times (n+1) \times n$. Further, because n is prime and not a factor of $k!$, we can remove n from the right part and it is done. □

Proof of Theorem 2.11. Assume f is a Latin matching from $\binom{[2n-1]}{n-1}$ to $\binom{[2n-1]}{n}$. Define f^+ as Definition 2.4. Assume k is any integer such that $2 \leq k \leq n-1$. Let S be the set of k-permutations of $[n+k]$. Formally,

$$S = \{(a_1, \dots, a_k) \mid a_i \in [n+k], a_i \neq a_j \text{ for } i \neq j\}.$$

Define a mapping $g : S \to [n+k]$ by

$$g(a_1, \dots, a_k) = f^+(\{a_1, \dots, a_k, n+k+1, n+k+2, \dots, 2n-1\}).$$

We claim that for any $i \in [n+k]$, there are exactly $|S|/(n+k)$ elements of S mapped to i under g. Intuitively speaking, g uniformly partitions S.

Let $\vec{b} = (b_1, \dots, b_{k-1})$ be a $(k-1)$-permutation of $[n+k]$. Denote by $\vec{b}_t$ the shifted sequence $(b_1+t, \dots, b_{k-1}+t)$, where additions are taken modulo $[n+k]$. Denote the union of elements in $\vec{b}_t$ by B_t, and denote B_0 as B for short. Using the definition of g and the Latin property of f, the value of $g(\vec{b}, x)$ runs over $[n+k] - B$ when x runs over $[n+k] - B$. Moreover, when t runs over $[n+k]$, the (multiset) union of B_t is clearly $(k-1)$ copies of $[n+k]$. Together, given $\vec{b}$, multiset $\{g(\vec{b}_t, x) \mid t \in [n+k], x \in [n+k] - B_t\} = \{[n+k] - B_t \mid t \in [n+k]\}$ equals $(n+k) - (k-1) = (n+1)$ copies of $[n+k]$. Furthermore, applying the fact that S can be partitioned as $\dot{\bigcup}_{b_1=1, b_2, \dots, b_{k-1}, t \in [n+k], x \in [n+k] - B_t} \{(\vec{b}_t, x)\}$, multiset $g(S)$ equals several copies of $[n+k]$, i.e. g uniformly partitions S.

In particular, the number of elements in S that are mapped to 1 under g is $|S|/(n+k)$. However, this number should be a multiple of $k!$ since g is invariant under permutation. Therefore, $k!$ divides $|S|/(n+k) = (n+k-1)\dots(n+1)$. Finally, applying Proposition 2.12, n must be prime. □

2.4 Short Summary & Additional Results

Table 1 summarizes our knowledge on Latin matchings.

order n	$2,3,5$	$7,11$	prime > 11	composite
Cyclic Latin	Yes	No	Unknown	(No)
Latin	(Yes)	Unknown	Unknown	No

Table 1: On the existences of Latin matchings

Thus far in this section we discuss the special case where $k=1, d=n-1$. The following theorem states some results for two other cases (proof will be given in a full version).

An ordered designs with parameters t, v, k, written as $OD(t,k,v)$, is an $k \times \binom{k}{t} t!$ array with v entries such that each column has k distinct entries and any t rows contain each column tuple of t distinct entries exactly once [8, 10, 23].

Theorem 2.13. (1) *$p'_{\max} = 1/\binom{k+d}{d}$ when $n=2$.*

(2) *For $k=1$, equality $p'_{\max} = 1/\binom{k+d}{d} = 1/(d+1)$ holds if and only if there exists an ordered design $OD(n-1, n, n+d)$ (because an $OD(n-1,n,n+d)$ corresponds to an independent set of $G(n,1,d)$ of size $s/(d+1)$).*

Remark. *1. Since $OD(n-1,n,n+1)$ (trivially) exists [10], the aforementioned equality of $p'_{\max}$ holds when $k=d=1$.*

2. When n is prime, by [25] there is an $OD(n-2, n-1, 2n-2)$. Thus the case $(n-1,1,n-1)$ is solved for any prime n. However, it is open whether an $OD(n-1,n,2n-1)$ exists. If so, $(n,1,n-1)$ is solved for prime n. (Note that an $OD(n-1,n,2n-1)$ implies an $OD(n-2,n-1,2n-2)$.)

Why do we focus on $k=1, d=n-1$? The all-right case is quite difficult; so we fix $k=1$. When $k=1$, it seems that $d=1$ or $d=n-1$ is relatively easy. Another reason is that from this typical case, it is enough to compare the difference between the two winning rules (as shown below).

3 ONE-RIGHT WINNING RULE

In this section, we move on to the one-right winning rule. Let $p_{\max}(n,k,d)$ (or $p_{\max}$ when (n,k,d) are clear) be the maximum winning probability for this rule. The next lemma gives a graph interpretation of the game in this rule.

Lemma 3.1. *Let $\nu(H)$ denote the cardinality of the maximum matching of graph H. Then $p_{\max} = \nu(H(n,k,d))/s$. Moreover, the best strategy can be easily computed provided that the maximum matching of $H(n,k,d)$ is given.*

Proof. Recall $H(n,k,d) = (U,V,E)$. A strategy can be represented by a subset E' of E in which each vertex in U has degree 1. Those vertices in V which are covered by edges of E' correspond to the winning placements of the strategy. This correspondence immediately implies the lemma. □

Theorem 3.2. $p_{\max} = \min\{1, n/\binom{k+d}{d}\}$.

Proof. Add a source x and a sink y to graph $H(n,k,d)$ as shown in Figure 1. Connect x to each vertex in U, and y to each vertex in V. Assume unit capacity for all edges. Clearly, $\nu(H(n,k,d)) = f$, where f denotes the value of maximum flow from x to y. By Lemma 3.1, $p_{\max} = \nu(H(n,k,d))/s$. Thus it reduces to proving that $f/s = \min\{1, n/\binom{k+d}{d}\}$.

When $n/\binom{k+d}{d} \le 1$, we need to prove $f = sn/\binom{k+d}{d} = nm$. First, $f \le \text{degree}(x) = nm$. To show $f \ge nm$, we construct a fractional flow with value nm: all edges connecting with x flow 1, all edges in E flow $1/\binom{k+d}{d}$, and all edges connecting with y flow $n/\binom{k+d}{d} \le 1$. The validity of flow is easy to check.

When $n/\binom{k+d}{d} \ge 1$, we need to prove $f = s$. First, $f \le \text{degree}(y) = s$. To show $f \ge s$, we construct the following fractional flow whose value is s: all edges connecting with y flow 1, all edges in E flow $1/n$, and all edges connecting with x flow $\binom{k+d}{d}/n \le 1$. The validity of flow is easy to check.

(Note that constructing a maximum fractional flow is sufficient here according to the *integral flow theorem* [27].) □

Remark. *Theorem 3.2 can be generalized to the asymmetric case where the dealer places different number of hats to different players and the players guess multiple and possibly different times. (Still, they win if at least one guess of a player is right.) Assume player i receive k_i hats and has to answer g_i times. Then, $p_{\max} \le \sum_{i=1}^{n} \frac{g_i}{\binom{k_i+d}{d}}$ by union bound. Surprisingly, by our flow technique, $p_{\max} = \min\{1, \sum_{i=1}^{n} \frac{g_i}{\binom{k_i+d}{d}}\}$. We leave the proof as an exercise for curious readers.*

The fractional flow technique can solve other guessing problems. Assume three players want to guess the order of four distinct numbers A, B, C, D. However, player 1 can only observe the order between B, C, player 2 can observe the order between A, B, C, whereas other two players can observe nothing. No communication is allowed after the observation and the rule is still one-right. Then, we can immediately claim that $p_{\max} = 1/12 + 1/4 + 1/24 + 1/24 = 5/12$.

The maximum winning probability (under one-right rule) is already solved by Theorem 3.2, whereas by Lemma 3.1 computing the best strategies is also solved (in general), which reduces to computing the maximum matching of $H(n,k,d)$, whose running time is a polynomial of the size of $H(n,k,d)$. However, to play the hat guessing game using such matchings is unpractical, because players have to remember the entire matching, which is enormous when n, k, d grow bigger.

Therefore, in the rest of this section, we aim to design cooperative strategies using codes. We first review two kinds of well-known matchings in the subset lattice, and then by utilizing such explicit matchings, we design **explicit** best cooperative strategies for the cases where $n=2$ or $k=1$.

3.1 Two Kinds of Explicit Matchings

In the following, we introduce a perfect matching γ_j^T between two antipodal layers $\mathcal{P}_j$ and $\mathcal{P}_{T-j}$ for each $j < \frac{T}{2}$, as well as a matching ϕ_j^T between two consecutive layers $\mathcal{P}_j$ and $\mathcal{P}_{j+1}$, which is injective when $j < \frac{T}{2}$ and surjective otherwise. Note: if A is matched to B in these matchings, it must hold $A \subset B$.

Let CW, CCW be short for clockwise and counterclockwise.

*Definition 3.3 (**CCW-rotating-subset**).* Assume $A \subset [T]$ and $|A| \leq T/2$. By two steps we define a subset $\circlearrowleft(A) = A'$ with equal size as A and is disjoint with A.

Step 1. Put $1, \ldots, T$ in CW into a cycle.

Step 2. Enumerate each number a in A, find the CCW first number near a that is not in $A \cup A'$ and add it to A'.

Note: the order of this enumeration does not matter. Take $T = 10$ and $A = \{1, 3, 8, 9\}$ for example. If we enumerate in order $1, 3, 8, 9$, the numbers added to A' would be $10, 2, 7, 6$. Alternatively, if we enumerate in order $3, 9, 8, 1$, the numbers added to A' would also be $2, 7, 6, 10$.)

Definition 3.4. Assume $j < T/2$. For any subset A of $[T]$ with size j, define $\gamma_j^T(A) = [T] - \circlearrowleft(A)$.

We omit the superscript T in γ_j^T when T is clear.

Apparently, γ_j is a perfect matching between $\mathcal{P}_j$ and $\mathcal{P}_{T-j}$.

We define the CW-rotating-subset $\circlearrowright(A)$ symmetrically and let $\gamma'_j(A) := [T] - \circlearrowright(A)$ for $j < T/2$. Note that $\circlearrowright(\cdot)$ is the reverse function of $\circlearrowleft(\cdot)$; namely, $\circlearrowright(\circlearrowleft(A)) = A$.

We now introduce a matching ϕ_j^T between $\mathcal{P}_j$ and $\mathcal{P}_{j+1}$. To this end we distinguish two cases as shown below.

Definition 3.5 (Case1: $j < T/2$). Consider an example where $j = 1, T = 4$. To define $\phi_1^4(A)$ from $\binom{[4]}{1}$ to $\binom{[4]}{2}$, we borrow $\gamma_2^5 : \binom{[5]}{2} \to \binom{[5]}{3}$ – first add 5 to A and compute $\gamma_2^5(A)$, then remove 5 from $\gamma_2^5(A)$. See the top of Figure 6.

In general, let $c = (T - (j+1)) - j = T - 2j - 1$, and $C = \{T+1, \ldots, T+c\}$. For $A \subset [T]$ and $|A| = j$, define

$$\phi_j^T(A) = \gamma_{j+c=T-j-1}^{T+c=2T-2j-1}(A \cup C) - C. \tag{2}$$

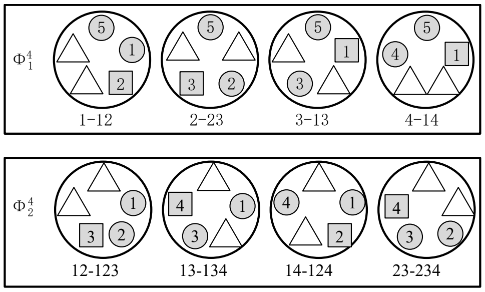

Figure 6: Illustration of the definitions of ϕ_1^4 and ϕ_2^4.

Definition 3.6 (Case 2: $j \geq T/2$). Consider an example where $j = 2, T = 4$. To define $\phi_2^4(A)$ from $\binom{[4]}{2}$ to $\binom{[4]}{3}$, we also borrow $\gamma_2^5 : \binom{[5]}{2} \to \binom{[5]}{3}$ – If $\gamma_2^5(A)$ does not contain 5, then it is defined to be $\phi_2^4(A)$. Clearly, $\phi_2^4(A)$ will be defined for only $\binom{4}{3}$ choices of A. See the bottom of Figure 6.

In general, Let $c = j - (T - (j+1)) = 2j + 1 - T$, and $C = \{T+1, \ldots, T+c\}$. For $A \subset [T]$ and $|A| = j$, define

$$\phi_j^T(A) = \begin{cases} \gamma_j^{T+c=2j+1}(A), & \text{if } \gamma_j^{2j+1}(A) \cap C = \varnothing; \\ \text{undefined}, & \text{otherwise}. \end{cases} \tag{3}$$

Lemma 3.7. *ϕ_j^T is a matching. Moreover, it is injective when $j < T/2$ and surjective when $j \geq T/2$. (Proof omitted.)*

To get better understanding of $\{\gamma_j^T\}$ and $\{\phi_j^T\}$, the reader can find their alternative definitions in subsection 3.3.

3.2 Explicit Strategies for One-right Rule

We now apply $\{\gamma_j^T, \phi_j^T\}$ to construct explicit strategies.

I. $n = 2$. *When Player 1 observes set B, she guesses $\circlearrowleft(B)$. When player 2 observes set A, he guesses $\circlearrowleft(A)$.* To prove its optimality, we shall prove that players never guess right simultaneously. Observe that γ_j, γ'_j are two perfect matchings that are disjoint; namely, $\circlearrowleft(A) \neq \circlearrowright(A)$ for any A such that $|A| < T/2$. When player 1 guesses right, we have $A = \circlearrowleft(B)$, so $B = \circlearrowright(A)$. This means $B \neq \circlearrowleft(A)$, so player 2 guesses wrong.

II. $k = 1, d = n - 1$. Let A denote the set of n hats of the players. Let A_i denote the set of hats on all players except player i. *After observing A_i, player i answers the 1-element set $\gamma_{n-1}(A_i) - A_i$.* Since γ_{n-1} is a perfect matching between $\mathcal{P}_{n-1}$ and $\mathcal{P}_n$, there exists one and only one element $a \in A$ such that $\gamma_{n-1}(A - \{a\}) = A$. Apparently, the player whom is placed by a is the only one who guesses right. So, exact one player is correct.

II'. $k = 1$. Let A_i and A be the same as II. *Player i answers the 1-element set in $\phi_{n-1}^T(A_i) - A_i$.* Note that player i can answer arbitrarily when $\phi_{n-1}^T(A_i)$ is undefined (which may happen when $n - 1 \geq \frac{T}{2} = \frac{n+d}{2}$; namely, when $d \leq n - 2$). When $d \geq n - 1$, at most one player would guess right since ϕ_{n-1}^T is injective (by Lemma 3.7), so this strategy achieves a winning probability of $n/(d+1)$ which is optimum. When $d \leq n - 1$, the winning probability is 1 because ϕ_{n-1}^T is surjective.

Observe that all the strategies above have two features: First, the players are unbiased – they use the same (individual) strategy to guess. Second, when applying such strategies, the players can put on masks so that no one recognized the identities of the others – as a player only needs to observe the set of hats assigned to the other players but not the full information including how the hats are distributed.

Such features in general are not enjoyed by the strategies computing from the network flow technique given with Theorem 3.2. However, using similar techniques we find that for any value of (n, k, d) there always exists an optimal strategy that enjoys the above two features (for one-right rule). This result will be reported in the full version of this paper.

Also, observe that if γ_{n-1} is replaced by a Latin matching of order n in our strategy for the case $(k = 1, d = n - 1)$, another good feature can be proved (while the two features above still hold): In every possible assignment of the hats, in addition to the fact that exactly one player is right, the $n-1$ incorrect players would guess $n-1$ different colors which are exactly the $d = n-1$ colors remain in dealer's hand.

3.3 A Chain Decomposition of the Subset Lattice and Its Connections with γ, ϕ

Recall the matchings $\{\gamma\}$ and $\{\phi\}$ given in subsection 3.1. We point out that in literature there are equivalent definitions for these matchings. We review these definitions in this subsection to help the reader to know better of $\{\gamma\}$ and $\{\phi\}$.

Parenthesis sequence of a set. Given $A \subseteq [T]$. We can associate with A a sequence of parenthesis of length T. First, write down numbers $1, \ldots, T$ in order. If $x \in A$, replace x by a right parenthesis; otherwise, replace x by a left parenthesis. For example, if $T = 10$, the sequence associated with $A = \{1,3,4,8,9\}$ is $)_1 \quad (_2 \quad)_3 \quad)_4 \quad (_5 \quad (_6 \quad (_7 \quad)_8 \quad)_9 \quad (_{10}$. This sequence of parenthesis can be parenthesized uniquely in the usual way, and there may remain several parenthesis unpaired. For the example, "$(_2,(_6,(_7$" are paired with "$)_3,)_9,)_8$". All the others are unpaired. Clearly, any unpaired right parenthesis occur to the left of any unpaired left parenthesis.

Definition of λ_j^T [15]. Assume A is a j-element subset of $[T]$ associated with sequence S. Replace the leftmost unpaired '(' in S by ')' and assume that the new sequence corresponds to subset A', then $\lambda_j^T(A)$ is defined as A'. For the above example, the leftmost unpaired '(' is $(_5$, so $\lambda_5^{10}(A) = \{1,3,4,5,8,9\}$.

The functions $\lambda_0^T, \ldots, \lambda_{T-1}^T$ lead to a chain-decomposition of the subset lattice of $[T]$ [15]. We regard that two subsets of $[T]$ are in the same *chain*, if and only if their associated sequences contain the same paired parenthesis. The entire chain containing $\{1,3,4,8,9\}$ is $\{3,8,9\}$ - $\{1,3,8,9\}$ - $\{1,3,4,8,9\}$ - $\{1,3,4,5,8,9\}$ - $\{1,3,4,5,8,9,10\}$. The entire chain-decomposition for $T = 4$ is depicted as follows.

$$\begin{array}{lllllllll}
\{\} \to & \{1\} & \to & \{1,2\} & \to & \{1,2,3\} & \to \{1,2,3,4\} \\
 & \{2\} & \to & \{2,3\} & \to & \{2,3,4\} & \\
 & \{3\} & \to & \{1,3\} & \to & \{1,3,4\} & \\
 & \{4\} & \to & \{1,4\} & \to & \{1,2,4\} & \\
 & & & \{2,4\} & & & \\
 & & & \{3,4\} & & &
\end{array}$$

Definition of β_j^T [15]. All chains in the above decomposition are *symmetric* - if a chain contains some member A, it must contain a member with size $T - |A|$. Therefore, for each $j < \frac{T}{2}$, this chain-decomposition implicitly defines an antipodal matching β_j between the two antipodal layers $\mathcal{P}_j$ and $\mathcal{P}_{T-j}$.

Lemma 3.8. *[17] 1. $\beta_j = \gamma_j$. 2. $\lambda_j^T = \phi_j^T$.*

By Lemma 3.8, subsection 3.1 in fact provides new equivalent definitions of the well-known matchings λ and β. Our definition of β (i.e. γ) however seems more intuitive. Our method for defining λ (i.e. ϕ) is original, which borrows the matching γ. In contrast, the authors in [15] first defined the matchings between any two consecutive layers (λ) and then used them to define the antipodal matchings (β). Also, note that [2] provides another equivalent definition of λ.

4 CONCLUDING REMARKS

We study an interesting variant of the hat guessing game and show its strong connections to several combinatorial structures such as ordered designs, antipodal matchings, and Latin Matchings. We summarize our results in Table 2.

Rule	**Max winning probability**	**Cases with an explicit optimal strategy**
One-right	$\min\{1, n/\binom{k+d}{d}\}$	$n = 2$ or $k = 1$
All-right	smaller than or equal to $1/\binom{k+d}{d}$	$n = 2$ or ($k = 1$ & $\exists OD_1(n-1, n, n+d)$)

Table 2: Summary of the main results

Comparing our unbiased strategy (where everybody use the same strategy for guessing) for one-right rule with our almost-unbiased strategy (where there is a leader and the other players are unbiased) for all-right rule under the case $(n, 1, n-1)$, we observe an interesting phenomenon that a leader / dictator might be necessary for making consensus but is unnecessary for decentralization in team cooperation.

Some applications of these strategies are already mentioned in the remark of subsection 2.2, while another is as follows. Since the strategy to our variant has the same favor as the known strategies (e.g. Hamming codes) to the original version, they fit in a similar *de-randomization* technique for protocols or auctions. Take $(n, k, d) = (n, 1, n-1)$ for an example. Clearly, the number of correct guesses is 1 in expectation, since each player has $1/n$ chance to be right. Using our one-right strategy, this becomes deterministic 1. Using our all-right strategy, we can design an n-round deterministic strategy so that all right guesses happen in the same round.

The Latin matching has rich structural properties and may find applications in *factorial experiments*, *block cipher design*, *secret sharing* (see such applications of similar structures in [16]) and also in other combinatorial designs. Following Lemma 2.9, our construction of Latin matchings leads to the construction of the large set of $OD(n-1, n, 2n-1)$ (denoted by $LOD(n-1, n, 2n-1)$ [10]). Thus we obtain $LOD(1,2,3), LOD(2,3,5), LOD(4,5,9)$ as byproducts. In particular, $OD(4,5,9)$ and $LOD(4,5,9)$ are the first time reported. A curious yet challenging open problem is how to design $OD(6,7,13)$ or $LOD(6,7,13)$, and in general, $OD(n-1, n, 2n-1)$ for prime n larger than 5. We hope this paper could stimulate the ongoing research on ordered designs.

To the best of our knowledge, Latin matchings have not been reported in literature. Exploring other such matchings is important in further study. At the end of this paper, we note that the Latin matching depicted in Figure 4 enjoys some symmetries: if A is mapped to B, $[9]/B$ is mapped to $[9]/A$ and $-A$ is mapped to $-B$, where $-A := \{9 - a \mid a \in A\}$.

ACKNOWLEDGMENTS

We thank Professor Xiaoming Sun and Professor Zhiyi Huang for taking part in some discussions.

REFERENCES

[1] G. Aggarwal, A. Fiat, A. V. Goldberg, J. D. Hartline, N. Immorlica, and M. Sudan. 2011. Derandomization of auctions. *Games and Economic Behavior* 72, 1 (2011), 1 – 11. https://doi.org/10.1016/j.geb.2010.07.007

[2] M. Aigner. 1973. Lexicographic matching in Boolean algebras. *Journal of Combinatorial Theory, Series B* 14, 3 (1973), 187 – 194. https://doi.org/10.1016/0095-8956(73)90001-4

[3] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. 1991. The Expressive Power of Voting Polynomials. In *Proceedings of the 23rd ACM Sympo. on Theory of Computing*. ACM, 402–409. https://doi.org/10.1145/103418.103461

[4] O. Ben-Zwi, I. Newman, and G. Wolfovitz. 2015. Hats, Auctions and Derandomization. *Random Structures & Algorithms* 46, 3 (2015), 478–493. https://doi.org/10.1002/rsa.20512

[5] T. Beth, D. Jungnickel, and H. Lenz. 1999. *Design Theory*. Cambridge University Press.

[6] E. Brown and J. Tanton. 2009. A Dozen Hat Problems. (2009). https://www.math.vt.edu/people/brown/doc/dozen

[7] S. Butler, M.T. Hajiaghayi, R.D. Kleinberg, and T. Leighton. 2009. Hat Guessing Games. *SIAM Rev.* 51, 2 (2009), 399–413. https://doi.org/10.1137/080743470

[8] C.J. Colbourn and J.H. Dinitz (Eds.). 1996. *CRC Handbook of Combinatorial Designs*. CRC Press, Inc.

[9] Charles J. Colbourn and Paul C. van Oorschot. 1989. Applications of Combinatorial Designs in Computer Science. *ACM Comput. Surv.* 21, 2 (1989), 223–250. https://doi.org/10.1145/66443.66446

[10] J.H. Dinitz and D.R. Stinson (Eds.). 1992. *Contemporary Design Theory: A Collection of Surveys*. Wiley-Interscience.

[11] T. Ebert, W. Merkle, and H. Vollmer. 2003. On the Autoreducibility of Random Sequences. *SIAM J. on Computing* 32, 6 (2003), 1542–1569. https://doi.org/10.1137/S0097539702415317

[12] U. Feige. 2004. *You Can Leave Your Hat On (if you guess its color)*. Technical Report. The Weizmann Institute of Science. http://www.wisdom.weizmann.ac.il/~feige/TechnicalReports/hats.ps

[13] M. Gadouleau and N. Georgiou. 2015. New Constructions and Bounds for Winkler's Hat Game. *SIAM J. on Discrete Mathematics* 29, 2 (2015), 823–834. https://doi.org/10.1137/130944680

[14] M. Gadouleau and S. Riis. 2011. Graph-Theoretical Constructions for Graph Entropy and Network Coding Based Communications. *IEEE Trans. on Infor. Theory* 57, 10 (Oct 2011), 6703–6717. https://doi.org/10.1109/TIT.2011.2155618

[15] C. Greene and D. J Kleitman. 1976. Strong Versions of Sperner's Theorem. *J. of Comb. Theory, Series A* 20, 1 (1976), 80 – 88. https://doi.org/10.1016/0097-3165(76)90079-0

[16] A.S. Hedayat, N.J.A. Sloane, and J. Stufken. 1999. *Orthogonal Arrays Theory and Applications*. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-1478-6

[17] K. Jin. 2017. Toward 1-factorizations of Bipartite Kneser Graphs. (2017). arXiv:1704.08852 http://arxiv.org/abs/1704.08852

[18] K. Jin. 2019. CyclicLatin7-in-HAT-forAAMAS19. https://github.com/cscjjk/CyclicLatin7-in-HAT-forAAMAS19. (2019).

[19] K. Jin. 2019. MISofG-4-1-3-inHAT-forAAMAS19. https://github.com/cscjjk/MISofG-4-1-3-inHAT-forAAMAS19. (2019).

[20] H.W.J. Lenstra and G. Seroussi. 2002. On Hats and Other Covers. In *Proc. IEEE Inter. Sympo. on Information Theory*. IEEE, 342–342. https://doi.org/10.1109/ISIT.2002.1023614

[21] T. Ma, X. Sun, and H. Yu. 2011. A New Variation of Hat Guessing Games. In *Computing and Combinatorics*. Springer Berlin Heidelberg, 616–626. https://doi.org/10.1007/978-3-642-22685-4_53

[22] B.D. McKay and I.M. Wanless. 2008. A Census of Small Latin Hypercubes. *SIAM J. Discrete Math.* 22 (03 2008), 719–736. https://doi.org/10.1137/070693874

[23] C.R. Rao. 1961. Combinatorial Arrangements Analogous to Orthogonal Arrays. *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)* 23, 3 (1961), 283–286.

[24] A. Rosa. 1980. *Topics on Steiner systems*. North-Holland Publishing Company.

[25] L. Teirlinck. 1988. On large set of disjoint ordered design. *Ars Combinatoria* 17 (1988), 31–37.

[26] The-New-York-Time. 2001. Why Mathematicians Now Care About Their Hat Color. http://www.nytimes.com/2001/04/10/science/why-mathematicians-now-care-about-their-hat-color.html. (2001).

[27] Wikipedia. 2018. Maximum flow problem. https://en.wikipedia.org/wiki/Maximum_flow_problem. (2018).