SIAM J. DISCRETE MATH.
Vol. 28, No. 3, pp. 1578–1600

MUSICAL CHAIRS*

YEHUDA AFEK†, YAKOV BABICHENKO‡, URIEL FEIGE§, ELI GAFNI¶, NATI LINIAL‖, AND BENNY SUDAKOV#

Abstract. In the *musical chairs* game $MC(n, m)$, a team of n players plays against an adversarial *scheduler*. The scheduler wins if the game proceeds indefinitely, while termination after a finite number of rounds is declared a win of the team. At each round of the game each player *occupies* one of the m available *chairs*. Termination (and a win of the team) is declared as soon as each player occupies a unique chair. Two players that simultaneously occupy the same chair are said to be *in conflict*. In other words, termination (and a win for the team) is reached as soon as there are no conflicts. The only means of communication throughout the game is this: At every round of the game, the scheduler selects an arbitrary nonempty set of players who are currently in conflict, and notifies each of them separately that it must move. A player who is thus notified changes its chair according to its deterministic program. As we show, for $m \geq 2n - 1$ chairs the team has a winning strategy. Moreover, using topological arguments we show that this bound is tight. For $m \leq 2n - 2$ the scheduler has a strategy that is guaranteed to make the game continue indefinitely and thus win. We also have some results on additional interesting questions. For example, if $m \geq 2n - 1$ (so that the team can win), how quickly can they achieve victory?

Key words. distributed computing, oblivious computing, renaming, probabilistic analysis, asynchronous computation

AMS subject classifications. 68Q02, 91A02

DOI. 10.1137/12088478X

1. Introduction.

Communication is a crucial ingredient in every kind of collaborative work. But what is the least possible amount of communication required for a team of players to achieve certain goals? Motivated by this question, we consider in this paper the following game.

The *musical chairs* game $MC(n, m)$ involves m *chairs* numbered $1, \ldots, m$ and a team of n *players* $P_1, \ldots, P_n$, who are playing against an adversarial *scheduler*. The scheduler's goal is to make the game run indefinitely, in which case it wins. The termination condition is that each player settles in a different chair. Upon termination the team of players is declared the winner. We say that player P is *in conflict* if some other player Q is presently occupying the same chair as P. Namely, termination and a win of the team is reached if there are no conflicts. The scheduler gets to decide for each player P_i the chair that P_i occupies at the start of the game. As mentioned above we consider here the *oblivious* model of communication which severely restricts

*Received by the editors July 16, 2012; accepted for publication (in revised form) July 11, 2014; published electronically September 25, 2014.

http://www.siam.org/journals/sidma/28-3/88478.html

†The Blavatnik School of Computer Science, Tel-Aviv University, Israel 69978 (afek@tau.ac.il).

‡Department of Mathematics, Hebrew University, Jerusalem 91904, Israel (yak@math.huji.ac.il).

§Department of Computer Science and Applied Mathematics Weizmann Institute of Science Rehovot 76100, Israel (uriel.feige@weizmann.ac.il). The author holds the Lawrence G. Horowitz Professorial Chair at the Weizmann Institute. Research supported in part by the Israel Science Foundation (grant 873/08).

¶Computer Science Department, University of California, Los Angeles, CA 95024 (eli@cs.ucla.edu).

‖School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel (nati@cs.huji.ac.il).

#Department of Mathematics, ETH, 8092 Zurich, Switzerland, and Department of Mathematics, UCLA, Los Angeles, CA 90095 (benjamin.sudakov@math.ethz.ch). This author's research was supported in part by SNSF grant 200021-149111 and by a USA-Israel BSF grant.

the communication between the players during the game. All such communication is mediated by the scheduler as follows: At every time step, and as long as there are conflicts, the scheduler selects an arbitrary nonempty set of players which are currently in conflict and notifies them that they need to move. (Later on we consider various restrictions on the selected sets resulting in different schedulers; see section 2.) A player thus notified to be in conflict changes its chair according to its deterministic program, which the player chooses before the game. During the game, each player has no information about the chairs of other players beyond the occasional one bit that tells it that it must move, and we insist that the choice of a player's next chair be deterministic. Consequently, a player's action depends only on its current chair and the sequence of chairs that it had traversed so far. Therefore the sequence of chairs that player P_i traverses is simply an infinite word π_i over the alphabet $1, \ldots, m$. Recall that the adversary can start each player on any of the chairs. Consequently we assume that each π_i is *full*, i.e., it contains all the letters in $[m]$. So upon receiving a conflict notification from the scheduler, player P_i occupying chair $\pi_i[k]$ moves to chair $\pi_i[k+1]$. The scheduler's freedom in choosing the players' initial chairs means that for every i the scheduler selects an index k_i and the game starts with each P_i occupying chair $\pi_i[k_i]$. A winning strategy for the players is a choice of full words π_i with the following property: For every choice of initial positions k_i and for every strategy of the scheduler the game terminates in a finite number of rounds, i.e., the players cannot be beaten by the scheduler.

In this paper we obtain several results about the musical chairs game. The reader should note that as discussed in [3] the musical chairs problem is strongly related to the adaptive renaming (also known as strong renaming) problem [1, 10]. Our first theorem determines the minimal m for which the team of players wins the $MC(n, m)$ game.

THEOREM 1. *The team of players has a winning strategy in the $MC(n, m)$ game iff $m \geq 2n - 1$.*

In the winning strategy that we produce, each word π_i is *periodic*, or, equivalently, a finite word that P_i traverses cyclically. We also show that for every $N > n$ there exist N full cyclic words on the alphabet $[m] = [2n - 1]$ such that every set of n out of these N words constitutes a winning strategy for the $MC(n, 2n - 1)$ game.

To prove the lower bound in Theorem 1 we use Sperner's lemma (see, e.g., [4]), a fundamental tool from combinatorial topology. The use of this tool in proving lower bounds for distributed algorithms was pioneered in [7, 10, 11]. If one is willing to assume a great deal of knowledge in the field of distributed computing, it is possible to deduce our lower bound from known results in this area. Instead, to make the paper self contained, we chose to include here a direct proof which we think is more illuminating and somewhat simpler than the one which would result from reductions to the existing literature.

Although the words in Theorem 1 use the least number of chairs, namely, $m = 2n - 1$, their *lengths* are doubly exponential in n. This leads to several interesting questions. Are there winning strategies for the MC game with much shorter words, say, of length $O(n)$? Perhaps even of length m? Can we provide significantly better upper bounds on the number of rounds till termination? Even if the scheduler is bound to lose the game, how long can it make the game last? Our next two results give some answers to these questions. Here we consider an $MC(n, m)$ *winning systems* with N words. This is a collection of $N \geq n$ full words on $[m]$, every n of which constitute a winning strategy for the players in the $MC(n, m)$ game.

THEOREM 2. *For every $N \geq n$, almost every choice of N words of length $cn \log N$ in an alphabet of $m = 7n$ letters is an $MC(n, m)$ winning system with N full words.*

Moreover, every game on these words terminates in $O(n \log N)$ *steps. Here* c *is an absolute constant.*

Since we are dealing with full words that we seek to make short, we are ultimately led to consider the problem under the assumption that each (finite, cyclically traversed) word π_i is a *permutation* on $[m]$. We note that the context of distributed computing offers no particular reason for this restriction and that we are motivated to study this question due to its aesthetic appeal. We can design permutation-based winning strategies for $MC(n, 2n-1)$ game for very small n (provably for $n = 3$, a computer assisted construction and proof for $n = 4$). We suspect that no such constructions are possible for large values of n, but we are unable at present to show this. We do know, though, the following.

THEOREM 3. *For every integer* $d \geq 1$ *there is an* $MC(n, m)$ *winning system with* $N = n^d$ permutations *on* $m = cn$ *symbols, where* c *depends only on* d*. In fact, this holds for almost every choice of* N *permutations on* $[m]$*.*

We should stress that our proofs of Theorems 2 and 3 are purely existential. The explicit construction of such systems of words remains largely open, though we have the following result in this direction.

THEOREM 4. *For every integer* $d \geq 1$ *there is an* $MC(n, m)$ *winning system with* $N = n^d$ permutations *on* $m = O(d^2 n^2)$ *symbols.*

We conclude this introduction with a discussion of several additional aspects of the subject.

Our work was originally motivated by some questions in distributed computing. In every distributed algorithm each processor must occasionally observe the activities of other processors. This can be done by reading the messages that other processors send, by inspecting some publicly accessible memory cells into which they write, or by sensing an effect on the environment due to the actions of other processors. Hence it is very natural to ask: What is the least possible amount of communication required to achieve certain goals? To answer it, we consider two severe limitations on the processors' behavior and ask how this affects the system's computational power. First, a processor can only post a proposal for its own output, and second, each processor is "blindfolded" and is only occasionally provided with the least possible amount of information, namely, a single bit that indicates whether its current state is "good" or "bad." Here bad/good indicates whether or not this state conflicts with the global-state desired by all the processors. Moreover, we also impose the requirement that algorithms are deterministic, i.e., use no randomization. This new minimalist model, which we call the oblivious model, was introduced in the conference version of this paper [3]. This model might appear to be significantly weaker than other (deterministic) models studied in distributed computing. Yet, our results show that a very natural distributed problem, *musical benches* [8], can be solved optimally within the highly limited oblivious model. Further discussion of the oblivious model and additional well-known problems like *adaptive renaming* [1, 2], which we can also be optimally solved in this model, can be found in [3].

A winning strategy for the $MC(n, m)$ game cannot include any two identical words. For that allows the scheduler to move the corresponding players together in lock-step, keeping them constantly in a state of conflict. Also for every winning strategy for $MC(n, m)$, with finite cyclic words, there is a finite upper bound on the number of moves till termination. To see this, let us associate with every state of the system a vector whose ith coordinate is the current position of player P_i on π_i. The set of such vectors V is finite, $|V| = \prod |\pi_i|$, and in a terminating sequence of moves no vector can be visited twice. In fact, we can associate with every collection of finite words a directed graph on vertex set V, where edges correspond to the possible transitions in response to scheduler's notifications. The collection of words constitute a winning MC strategy iff this directed graph is acyclic. We note that these observations depend on the assumption that players use no randomness.

Our strategies for the MC game have a number of additional desirable properties. As mentioned, we construct N full periodic words such that every subset of n of the N words constitutes an $MC(n, m)$ winning system. Hence our strategies are guaranteed to succeed (reach termination against every scheduler's strategy) in dynamic settings in which the set of players in the system keeps changing. This statement holds provided there are sufficiently long intervals throughout which the set of players remains unchanged. To illustrate this idea, consider a company that manufactures N communication devices, each of which can use any one of m frequencies. If several such devices happen to be at the same vicinity, and simultaneously transmit at the same frequency, then interference occurs. Devices can move in or out of the area, hop to a frequency of choice and transmit at this frequency, and sense whether there are other transmissions in this frequency. The company wants to provide the following guarantee: If no more than n devices reside in the same geographical area, then no device will suffer more than a total of T interference events for some guaranteed bound T. Our strategy for the MC game would yield this by preinstalling in each device a list of frequencies (a *word* in our terminology), and having the device hop to the next frequency on its list (in a cyclic fashion) in response to any interference it encounters. No communication beyond the ability to sense interference is needed.

In proving the lower bound $m \geq 2n-1$ we have to make several assumptions about the setup. The first is the freedom of choice for the scheduler. From the perspective of distributed computing this means that we are dealing with an asynchronous system. In a synchronous setting, in every time step, every player involved in a conflict moves to its next state. One can show that in such a synchronous setup the players have a winning strategy even with $m = n$ chairs. It is also important that the scheduler can dictate each player's starting position. If each P_i starts at the first letter of π_i, a trivial winning strategy with $m = n$ simply sets i as the first letter of π_i for each i. It is also crucial that our players are deterministic (no randomization). If players are allowed to pick their next state randomly, then again $m = n$ suffices, since in this case with probability 1 eventually a conflict-free configuration will be reached. Hence, this paper is also related to the one of the fascinating questions in computer science, whether and to what extent randomization increases the power of algorithmic procedures. Our results show that without using randomness one can still win an MC game by increasing only slightly the number of chairs (from n to $2n - 1$).

2. Simplified oblivious model for musical chairs. Our general model for oblivious algorithms is specified by providing the rules of possible behavior of the scheduler. Here we consider an *immediate scheduler*, who enjoys a high degree of freedom in choosing which processor to move.[1] To simplify the design and analysis of oblivious algorithms, it is convenient to consider a more restricted scheduler that has fewer degrees of freedom but is nevertheless equivalent to the immediate scheduler in their power to win the MC game. In each round an immediate scheduler can select an arbitrary nonempty set of players that are currently in conflict and move them. Below

[1] There are several interesting schedulers which are even more flexible than the immediate scheduler. As we showed in the conference version of this paper [3], all of them are, in fact, equivalent to the canonical scheduler.

we often refer to a team strategy as an oblivious $MC(n,m)$ algorithm. It is a winning strategy if the immediate scheduler is forced to reach a conflict-free configuration in finite time. Conversely, an immediate scheduler wins against an oblivious $MC(n,m)$ algorithm if it can generate an infinite execution without ever reaching a conflict-free configuration.

Terminology. Two schedulers σ_1, σ_2 are considered *equivalent* if for every team strategy, scheduler σ_1 has a winning strategy iff σ_2 has also. In other words, an oblivious $MC(n,m)$ algorithm beats σ_1 iff it beats σ_2.

First we want to limit the number of processors that can be moved in a round.

A *pairwise immediate scheduler* is similar to the immediate scheduler, except for the following restriction. In every round, the pairwise immediate scheduler can select any two processors $P \neq Q$ that are currently in conflict with each other and move either P, or Q, or both. Equivalently, in every round either exactly one processor (that is involved in a conflict) moves, or two processors that share the same chair move. We note that similar ideas can be found in the literature, Namely, that no generality is lost upon weakening the scheduler in an appropriate way, e.g., see [5, 7, 10, 11].

PROPOSITION 5. *The immediate scheduler and the pairwise immediate scheduler are equivalent.*

Proof. Clearly, every adversary of the pairwise immediate scheduler is also a strategy of the immediate scheduler. Hence it remains to show that if the immediate scheduler σ can win against some team strategy, then the pairwise immediate scheduler σ' can also force an infinite run against it. So fix an oblivious $MC(n,m)$ algorithm and an infinite run that σ forces. We modify this schedule into a schedule of σ' and show that every move that σ does is eventually performed by σ' as well. This is clearly, therefore, a winning strategy of σ'.

Consider the set X of the processors that σ moves at round t. The same set X will be moved by σ' as well, except that this is done in several steps. We split the set X according to the chair c that these processors occupied before round t. Thus, if σ moves exactly one of the ≥ 2 processors that occupy chair c, then σ' does just the same. If σ moves $k \geq 2$ processors that occupy the same chair c before round t, then σ' moves $k-2$ of these processors one by one and then finally moves the last two of these processors at once. The claim follows. ∎

The pairwise immediate scheduler (which, as we showed, is equivalent to the immediate scheduler) is instrumental in proving Theorems 1 and 3. However, to prove Theorem 2 the pairwise immediate scheduler needs to be further restricted. It is true that the pairwise immediate scheduler has to pick only one pair of players to move (and then either move only one or both of them), but it is still free to pick a pair of its choice (among those pairs that are in conflict). We would like to eliminate this degree of freedom.

Canonical scheduler. The canonical scheduler is similar to the pairwise immediate scheduler but with the following difference. In every round in which there is a conflict, one designates a *canonical pair.* This is a pair of players currently in conflict with each other, but they are not chosen by the scheduler, but rather dictated to the scheduler. Given the canonical pair P, Q, the only choice the scheduler has is whether to move P, or Q, or both. But how is the canonical pair chosen? It actually does not really matter, as long as the choice is deterministic. For concreteness, we shall assume the following procedure. Fix an arbitrary order on the collection of all pairs of players. In a nonterminal configuration, the *canonical pair* is the first pair of players in the order that share a chair.

PROPOSITION 6. *The canonical scheduler and the pairwise immediate scheduler are equivalent.*

Proof. The general nature of the proof is similar to that of Proposition 5, but the details are more complicated. Now σ is a pairwise immediate scheduler that achieves an infinite schedule S against some oblivious $MC(n,m)$ algorithm, and σ' is an arbitrary canonical scheduler. We need to construct an infinite schedule S' that σ' can accomplish against the same algorithm. We pass from S to S' through an infinite series of intermediate schedules S_t, each of which consists of a finite prefix λ_t followed by an infinite suffix ρ_t. We start with $S_0 = S$ whose prefix $\lambda_0 = \Lambda$ is empty. In passing from $S_t = \lambda_t \circ \rho_t$ to $S_{t+1} = \lambda_{t+1} \circ \rho_{t+1}$ we maintain the following invariants:

- For all t, the prefix λ_t of S_t is a canonical schedule.
- Each schedule S_t is an infinite pairwise immediate schedule.
- The prefix λ_t of S_t is a proper prefix of λ_{t+1}.

Note that this method of construction guarantees that for every position i in the emerging schedule there is some time T_i following which that position does not change. Consequently there is a well-defined limit to $S_0, S_1, \ldots$. This limit sequence is defined as the infinite schedule S'.

We turn to show how to construct S_{t+1} given S_t. Since S_t is infinite, the configuration at the end of λ_t is not terminal and there is a current canonical pair, say, P_1 and P_2, both of which presently occupy chair c_1. There are several cases to consider.

1. If the first move in ρ_t is canonical (i.e., all players moved at this step are in $\{P_1, P_2\}$). We simply take S_{t+1} to be S_t, thus satisfying the requirements stated above.
   We next turn to the cases where in the first move in ρ_t at least one player not from $\{P_1, P_2\}$ gets moved.
2. If either P_1 or P_2 is never moved in ρ_t. Let λ_{t+1} be λ_t followed by a move of P_2 (P_1). Let ρ_{t+1} be obtained from ρ_t by deleting the first move of P_2 (P_1) away from chair c_1 (if such move exists). Note that, as claimed, S_{t+1} is a pairwise immediate schedule. The only possible difficulty is that in some later stage in ρ_t some player P_3 is moved on account of its sharing chair c_1 with P_2 (P_1). But the same move of P_3 is realizable in S_{t+1} as well, since at this stage in S_{t+1} chair c_1 remains occupied by P_1 (P_2), so that P_3 is movable.
   We proceed to the cases where in ρ_t both P_1 and P_2 get moved away from c_1.
3. We now consider the case where in ρ_t the move of P_1 from c_1 precedes the first move of P_2 from c_1. Here we construct λ_{t+1} by appending a move of P_1 to λ_t, and ρ_{t+1} is obtained from ρ_t by deleting the first move of P_1 in ρ_t. All the requirements listed above are clearly satisfied.
   The same argument can be applied with P_1 and P_2 exchanged.
4. In the last case to consider, both players P_1 and P_2 are first moved out of c_1 at the same step s of ρ_t. There are two subcases to consider.
   - No player other than P_1 and P_2 is on chair c_1 in any of the first $s-1$ rounds of ρ_t. Here λ_{t+1} is λ_t followed by a move of both P_1 and P_2, and ρ_{t+1} is obtained by eliminating from ρ_t the moves of P_1, P_2 at round s.
   - In the complementary case some other player, say, P_3, occupies chair c_1 in round $s' < s$ and s' is the latest such time. We let λ_{t+1} be λ_t followed by a move of P_1. We obtain ρ_{t+1} by modifying ρ_t as follows: (i) Delete the move of P_1 at round s of ρ_t and (ii) move P_2 in ρ_{t+1} either together with P_3 (if only P_3 moves in round s'), or if P_3 moves together with a P_4, then move P_2 in the round immediately preceding the simultaneous move of P_3 and P_4. ∎

3. An oblivious MC algorithm with $2n-1$ chairs.

3.1. Preliminaries. In this section we prove the upper bound that is stated in Theorem 1. We start with some preliminaries. The length of a word w is denoted by $|w|$. The concatenation of words is denoted by $\circ$. The rth power of w is denoted by $w^r = w \circ w \ldots \circ w$ (r times). Given a word π and a letter c, we denote by $c \otimes \pi$ the word in which the letters are alternately c and a letter from π in consecutive order. For example, if $\pi = 2343$ and $c = 1$, then $c \otimes \pi = 12131413$. A collection of words $\pi_1, \pi_2, \ldots, \pi_n$ is called *terminal* if no schedule can fully traverse even one of the π_i. Every MC algorithm can be turned to a terminal collection by raising each word to a high enough power. Namely, given n words of length $\leq L$ each, if there is a schedule that runs for $> L^n$ steps, then there is necessarily a configuration that has been visited twice and therefore there is an infinite schedule as well. Therefore, it suffices to raise the words to L^n power to yield a terminal collection.

We now introduce some of our basic machinery in this area. We first show how to extend terminal sets of words.

PROPOSITION 7. *Let n, m, N be integers with $1 < n < m$. Let $\Pi = \{\pi_1, \ldots \pi_N\}$ be a collection of m-full words such that*

$$\text{(1)} \qquad \textit{every } n \textit{ of these words form an oblivious } MC(n,m) \textit{ algorithm.}$$

Then Π can be extended to a set of $N+1$ m-full words that satisfy condition (1).

Proof. Suppose that for every choice of n words from Π and for every initial configuration no schedule lasts more than t steps. As mentioned above $t \leq L^n$, where L is the length of the longest word in Π. For a word π, let π' be defined as follows: If $|\pi| \geq t$, then $\pi' = \pi$. Otherwise it consists of the first t letters in π^r, where $r > t/|\pi|$. The new word that we introduce is $\pi_{N+1} = \pi'_1 \circ \pi'_2 \circ \cdots \circ \pi'_n$. It is a full word, since it contains the full word π_1 as a subword.

We need to show that every set Π' of $n-1$ words from Π together with π_{N+1} constitute an oblivious $MC(n,m)$ algorithm. Observe that in any infinite schedule involving these words, there must be infinitely many moves on the word π_{N+1}. Otherwise, if the schedule remains on a letter c in π_{N+1} from some point on, replace π_{N+1} by an arbitrary word from $\Pi - \Pi'$ and stay put on the letter c in this word. This contradicts our assumption that Π satisfies condition (1). (Note that this word contains the letter c by our fullness assumption.) But π_{N+1} moves infinitely often, and it is a concatenation of n words whereas Π' contains only $n-1$ words. Therefore, eventually π_{N+1} must reach the beginning of a word π_α for some $\pi_\alpha \notin \Pi'$. From this point onward, π_{N+1} cannot proceed for t additional steps, contrary to our assumption. ∎

Note that by repeated application of Proposition 7, we can construct an arbitrarily large collection of m-full words that satisfy condition (1).

We next deal with the following situation: Suppose that $\pi_1, \pi_2, \ldots, \pi_m$ is a terminal collection, and we concatenate an arbitrary word σ to one of the words π_i. We show that by raising all words to a high enough power we again have a terminal collection in our hands.

LEMMA 8. *Let $\pi_1, \pi_2, \ldots, \pi_p$ be a terminal collection of full words over some alphabet. Let σ be an arbitrary full word over the same alphabet. Then the collection*

$$(\pi_1)^k, (\pi_2)^k, \ldots, (\pi_{i-1})^k, (\pi_i \circ \sigma)^2, (\pi_{i+1})^k, \ldots, (\pi_p)^k$$

is terminal as well for every $1 \leq i \leq p$ and every $k \geq |\pi_i| + |\sigma|$.

Proof. We split the run of any schedule on these words into *periods* through which we do not move along the word $(\pi_i \circ \sigma)^2$. We claim that throughout a single period we do not traverse a full copy of π_j in our progress along the word $(\pi_j)^k$. The argument is the same as in the proof of Proposition 7. By pasting all these periods together, we conclude that during a time interval in which we advance $\leq |\pi_i| + |\sigma| - 1$ positions along the word $(\pi_i \circ \sigma)^2$ every other word $(\pi_j)^k$ traverses at most $|\pi_i| + |\sigma| - 1$ copies of π_j. In particular, there is a whole π_j in the jth word in the collection that is never visited. If the schedule ends in this way, no word is fully traversed, and our claim holds.

So let us consider what happens when a schedule makes $\geq |\pi_i| + |\sigma|$ steps along the word $(\pi_i \circ \sigma)^2$. We must reach at some moment the start of π_i in our traversal of the word $(\pi_i \circ \sigma)^2$. But our underlying assumption implies that from here on, no entire copy of π_l (for $l = 1, \ldots, p$) is fully traversed. Thus, no word in this collection is fully traversed, as claimed. ◻

Lemma 8 yields immediately the following holds.

Corollary 9. *Let $\pi_1, \pi_2, \ldots, \pi_p$ be a terminal collection of full words over some alphabet, and let $\pi_{p+1}, \pi_{p+2}, \ldots, \pi_n$ be arbitrary full words over the same alphabet. Then the collection*

$$(\pi_1 \circ \pi_2 \circ \cdots \circ \pi_n)^2, (\pi_1)^k, (\pi_2)^k, \ldots, (\pi_{i-1})^k, (\pi_{i+1})^k, \ldots, (\pi_p)^k$$

is terminal as well. This holds for every $1 \leq i \leq p$ and $k \geq \sum_{i=1}^n |\pi_i|$.

This is a special case of Lemma 8 where $\sigma = \pi_{i+1} \circ \ldots \pi_n \circ \pi_1 \ldots \circ \pi_{i-1}$.

3.2. The $\mathbf{MC(n, 2n-1)}$ upper bound. The proof we present shows somewhat more than Theorem 1 says. A useful observation is that the scheduler can "trade" a player P for a chair c. Namely, the scheduler can keep P constantly on chair c and be able, in return, to move any other player past chair c. In other words, this effectively means the elimination of chair c from all other words. This suggests the following definition: If π is a word over alphabet C and $B \subseteq C$, we denote by $\pi(B)$ the word obtained from π by deleting from it the letters from $C \setminus B$.

Our construction is recursive. An inductive step should add one player (i.e., a word) and two chairs. We carry out this step in two installments: In the first we add a single chair, and in the second one we add a chair and a player. Both steps are accompanied by conditions that counter the abovementioned trading option.

Proposition 10. *For every integer $n \geq 1$*

- *there exist full words $s_1, s_2, \ldots, s_n$ over the alphabet $\{1, 2, \ldots, 2n-1\}$ such that $s_1(A), s_2(A), \ldots, s_p(A)$ is a terminal collection for every $p \leq n$, and every subset $A \subseteq \{1, 2, \ldots, 2n-1\}$ of cardinality $|A| = 2p - 1$,*
- *there exist full words $w_1, w_2, \ldots, w_n$ over alphabet $\{1, 2 \ldots, 2n\}$, such that $w_1(B), w_2(B), \ldots, w_p(B)$ is a terminal collection for every $p \leq n$, and every subset $B \subseteq \{1, 2, \ldots, 2n\}$ of cardinality $|B| = 2p - 1$.*

The words $s_1, s_2, \ldots, s_n$ in Proposition 10 constitute a terminal collection and are hence an oblivious $MC(n, 2n-1)$ algorithm that proves the upper bound part of Theorem 1. In the rest of this section we prove Proposition 10.

Proof. As mentioned, the proof is by induction on n. For $n = 1$, clearly $s_1 = 11$ and $w_1 = 1122$ satisfy the conditions.

In the induction step we use the existence of $s_1, s_2, \ldots, s_n$ to construct $w_1, w_2, \ldots, w_n$. Likewise the construction of $s_1, s_2, \ldots, s_{n+1}$ builds on the existence of $w_1, w_2, \ldots, w_n$.

The transition from $w_1, w_2, \ldots, w_n$ to $s_1, s_2, \ldots, s_{n+1}$. To simplify notation we assume that the words $w_1, w_2, \ldots, w_n$ in the alphabet $\{2, 3, \ldots, 2n+1\}$ (rather than $\{1, 2, \ldots, 2n\}$) satisfy the proposition. Let $k := \sum |w_i|$ and define

$$\begin{aligned} &s_1 := 1 \otimes ((w_1 \circ w_2 \circ \cdots \circ w_n)^{2(2n+1)}), \\ \forall i = 2, \ldots n+1 \quad &s_i := (w_{i-1})^{k(2n+1)} \circ 1. \end{aligned}$$

Fix a subset $A \subseteq \{1, 2, \ldots, 2n+1\}$ of cardinality $|A| = 2p - 1$ with $p \leq n+1$, and let us show that $s_1(A), s_2(A), \ldots, s_p(A)$ is a terminal collection. There are two cases to consider:

We first assume $1 \notin A$. This clearly implies that $p \leq n$ (or else $A = \{1, 2, \ldots, 2n+1\}$ and in particular $1 \in A$). In this case the collection is

$$\begin{aligned} &s_1(A) := ((w_1(A) \circ w_2(A) \circ \cdots \circ w_n(A))^{2(2n+1)}), \\ \forall i = 2, \ldots p \quad &s_i(A) := (w_{i-1}(A))^{k(2n+1)}. \end{aligned}$$

By the induction hypothesis, the collection $w_1(A), w_2(A), \ldots, w_{p-1}(A), w_p(A)$ is terminal. We apply Corollary 9 and conclude that

$$(w_1(A) \circ w_2(A) \circ \cdots \circ w_n(A))^2, (w_1(A))^k, (w_2(A))^k, \ldots, (w_{p-1}(A))^k$$

is terminal as well. But the s_i are obtained by taking $(2n+1)$th powers of these words, so that $s_1(A), s_2(A), \ldots, s_p(A)$ is terminal as needed.

We now consider what happens when $1 \in A$.

We define $F_1 := (w_1(A) \circ w_2(A) \circ \cdots \circ w_n(A))^2$ and for $j > 1$, let $F_j := (w_{j-1}(A))^k$. We refer to F_i as the ith block. In our construction each word has $2n+1$ blocks, ignoring chair 1.

At any moment throughout a schedule we denote by $\mathcal{O}_1$ the set of players in $\{P_2, P_3, \ldots, P_p\}$ that currently occupy chair 1. We show that during a period in which the set $\mathcal{O}_1$ remains unchanged, no player can traverse a whole block. The proof splits according to whether $\mathcal{O}_1$ is empty or not.

Assume first that $\mathcal{O}_1 \neq \emptyset$, and pick some $i > 1$ for which P_i occupies chair 1 during the current period. As long as $\mathcal{O}_1$ remains unchanged, P_i stays on chair 1, so the words that the other players repeatedly traverse are as follows: For P_1 it is

$$w_1(A \backslash \{1\}) \circ w_2(A \backslash \{1\}) \circ \cdots \circ w_n(A \backslash \{1\})$$

and for P_j with $p \geq j \neq i \geq 2$ it is

$$w_{j-1}(A \backslash \{1\}).$$

We now show that no player can traverse a whole block (as defined above). Observe that the collection $\{w_\nu(A \backslash \{1\}) | \nu = 1, \ldots, p-1\}$ (including, in particular the word $w_{i-1}(A \backslash \{1\})$) is terminal. This follows from the induction hypothesis, because $|A \backslash \{1\}| = 2p-2$, and because the property of being terminal is maintained under the insertion of new chairs into words. Applying Corollary 9 to this terminal collection implies that this collection of blocks is terminal as well.

We turn to consider the case $\mathcal{O}_1 = \emptyset$. In this case player 1 cannot advance from a none-1 chair to the next none-1 chair, since the two are separated by the presently unoccupied chair 1. We henceforth assume that player P_1 stays put on chair $c \neq 1$, but our considerations remain valid even if at some moment player P_1 moves to chair 1. (If this happens, it will necessarily stay there, since $\mathcal{O}_1 = \emptyset$.) We are in a situation where players $P_2, P_3, \dots, P_p$ traverse the words $w_1(A\backslash\{1,c\}), w_2(A\backslash\{1,c\}), \dots, w_{p-1}(A\backslash\{1,c\})$. (Chair c, which is occupied by player P_1, can be safely eliminated from these words.) But $|A\backslash\{1,c\}| = 2p-3$, so by the induction hypothesis no player can traverse a whole $w_i(A\backslash\{1,c\})$, so no player can traverse a whole block.

We just saw that during a period in which the set $\mathcal{O}_1$ remains unchanged, no player can traverse a whole block.

Finally, assume toward contradiction that P_j fully traverses s_j for some index j, and consider the first occurrence of such an event. It follows that P_j has traversed $2n+1$ blocks, so that the set $\mathcal{O}_1$ must have changed at least $2n+1$ times during the process. However, for $\mathcal{O}_1$ to change, some P_i must either move to or away from a 1-chair in s_i. But 1 occurs exactly once in s_i, so every P_i can account for at most two changes in $\mathcal{O}_1$, a contradiction.

The transition from $\boldsymbol{s_1, s_2, \dots, s_n}$ to $\boldsymbol{w_1, w_2, \dots, w_n}$. We assume that the words $s_1, s_2, \dots, s_n$ in the alphabet $\{2, 3, \dots, 2n\}$ satisfy the proposition. Let $k := \sum |s_i|$ and define

$$\begin{aligned} & w_1 := 1 \otimes ((s_1 \circ s_2 \circ \cdots \circ s_n)^{2(2n+1)}), \\ \forall i = 2, \dots, n \quad & w_i := (s_{i-1})^{k(2n+1)} \circ 1. \end{aligned}$$

Fix a subset $B \subseteq \{1, 2, \dots, 2n\}$ with $|B| = 2p-1$. Then

$$\begin{aligned} & w_1(B) = 1 \otimes ((s_1(B) \circ s_2(B) \circ \cdots \circ s_n(B))^{2(2n+1)}), \\ \forall i = 2, \dots, p \quad & w_i(B) = (s_{i-1}(B))^{k(2n+1)} \circ 1 \end{aligned}$$

are exactly the same as in the previous transition just by replacing s with w and A with B. (In this case the induction hypothesis is on s_i and we prove for w_i.) So exactly the same considerations prove that $w_1(B), w_2(B), \dots, w_m(B)$ is a terminal collection. $\square$

4. Impossibility results. In this section we prove the lower bound of Theorem 1. As it turns out, the situation for $2n-2 \geq m$ and for $m \geq 2n-1$ are dramatically different. As we saw, for $m \geq 2n-1$ the team has a winning strategy, but when $2n-2 \geq m$, not only is it true that the scheduler can win the game, but also the scheduler is guaranteed to have a winning strategy even if we (i) substantially relax the requirement that each word π_i over $[m]$ be full, or (ii) restrict its power to select the players' starting position on their words. In the next proposition case (i) occurs.

The following claim shows that the lower bound of Theorem 1 holds even for words which are not full, provided that they satisfy some "richness" condition specified herein. The specific condition is engineered so as to make it possible to apply our main topological tool, namely, Sperner's lemma.

PROPOSITION 11. *Every team strategy $\tau_1, \dots, \tau_n$ over $[m] = [2n-2]$ for which*

- *chair 1 appears in both τ_1, τ_2, and*
- *for every $3 \leq i \leq n$, the word τ_i contains both chair $2i-4$ and $2i-3$*

is a losing strategy.

Needless to say, this statement is invariant under permuting the player's names and the indices of the chairs. There are several such arbitrary choices of indices below, and we hope that this creates no confusion. In the impossibility results that we prove in this section, the number of chairs m is always $2n-2$. We also go beyond the lower bound of Theorem 1 by considering scenarios with a total of $N \geq n$ players and statements showing that there is a choice of n out of the N words that constitute a losing strategy. (Clearly, new words that get added to a losing team strategy make it only easier for the scheduler to win.) These deviations from the basic setup ($N \geq n$ words, weakened fullness conditions, starting points not controlled by the scheduler) give us more flexibility in our arguments and complement each other nicely. Here is one of the main theorems that we prove in this section. It yields exponentially many subsets of n words that constitute a losing team strategy.

THEOREM 12. *Let $N = 2n-2$ and let $\pi_1, \dots, \pi_N$ be words over $[m] = [2n-2]$ such that the only equality among the symbols $\pi_1[1], \pi_2[1], \pi_3[1], \dots, \pi_N[1]$ is $\pi_1[1] = \pi_2[1]$. Then, for every partition of the words $\pi_3, \dots, \pi_N$ into $n-2$ pairs, there is a choice of one word from each pair, such that the chosen words together with π_1 and π_2 constitute a losing team strategy even when the game starts on each word's first letter.*

Proposition 11 yields the lower bound of Theorem 1 under weakened fullness requirements. It is much less clear how Theorem 12 fits into the picture. We show next how to derive Proposition 11 from Theorem 12.

Proof (Theorem 12 implies Proposition 11). Let π_1 (resp., π_2) be the suffix of τ_1 (resp., τ_2) starting with the first appearance of the symbol 1. The other words come in pairs. For $3 \leq i \leq n$, we define π_{2i-3} to be the suffix of τ_i starting at chair $2i-4$, and π_{2i-2} is its suffix starting at chair $2i-3$. Theorem 12 implies that there is a choice of one word from each pair that together with π_1 and π_2 is losing when started from the initial chairs. The same scheduler strategy clearly wins the game on $\tau_1, \dots, \tau_n$ when started from the respective chairs. ∎

The proof of Theorem 12, which uses some simple topological methods, is presented in section 4.2. We provide all the necessary background material for this proof in section 4.1.

What happens if the fullness condition is eliminated altogether but the scheduler maintains its right to select the starting positions? The scheduler clearly loses against the words $\pi_i = (i)$ for $i = 1, \dots, m$. However, as the following theorem shows, once $N > m = 2n-2$, the scheduler has a winning strategy.

THEOREM 13. *For every collection of $N = 2n-1$ words over $[m] = [2n-2]$, there is a choice of n words and starting locations for which the scheduler wins.*

Proof. By the pigeonhole principle, the scheduler wins against every set of words S that together contain fewer than $|S|$ different letters. The whole collection satisfies this condition, since it has $2n-1$ words and only $\leq 2n-2$ letters. We consider such a collection of words S of smallest cardinality. By the minimality of $|S|$, the total number of letters that appear in the words of S is exactly $|S|-1$, otherwise just eliminate one word from S to get a smaller collection that satisfies this condition. If $|S| \leq n$, the scheduler can play against these $|S|$ players and win, as claimed. We remain with the case where $|S| > n$.

We create a bipartite graph one side of which consists of the words in S. The other side contains all the letters in S's words. There is an edge between vertex π and vertex x iff the letter x appears in the word π. Using the minimality of $|S|$ and applying the marriage theorem [9], we show that for every vertex that we remove from the words' side the remaining bipartite graph has a perfect matching. Stated differently, for every word $\pi \in S$ it is possible to mark one letter in every word in $S \setminus \{\pi\}$ where all the marked letters are different from each other. Let S' consist of π and the suffix of every other word in $S \setminus \{\pi\}$ starting from the marked letter. If $|S| = |S'|$ is even, then Theorem 12 applies since S' has more words than letters and there is exactly one coincidence among these words' initial letters. Consequently, S

has a subcollection of $\frac{|S|}{2} + 1 \le n$ that is a losing team strategy, as claimed. If $|S|$ is odd, we first delete a word from S whose marked letter differs from $\pi[1]$ and argue as above. Since the number of letters that appear in the words of S is $|S| - 1$, we will have after the deletion at least as many words as letters and can still apply Theorem 12. ☐

4.1. A few words on Sperner's lemma. In this section we discuss our main topological tool, Sperner's lemma (see, e.g., [4]). We include all the required background and try to keep our presentation to the minimum that is necessary for a proof of Theorem 12.

DEFINITION 14. *A* simplicial complex *is a collection X of subsets of a finite set of* vertices V *such that*

$$\textit{if} \;\; A \in X \;\; \textit{and} \;\; B \subseteq A, \;\; \textit{then} \;\; B \in X.$$

A member $A \in X$ is called a face*, and its* dimension *is defined as* $\dim A := |A| - 1$. *We refer to d-dimensional faces as d-faces, and define* $\dim X$ *as the largest dimension of a face in X. We note that a vertex is a 0-face, and call a 1-face an* edge*. The* 1-skeleton *of X is the graph with vertex set V, where xy is an edge iff $\{x, y\}$ is an edge (1-face) of X. A face of dimension* $\dim X$ *is called a* facet*. We say that X is* pure *if every face of X is contained in* some *facet. Finally, a* d-pseudomanifold *is a pure d-dimensional complex X such that*

(2) *every face of dimension $d - 1$ is contained in exactly two facets.*

A good simple example of a two-dimensional pseudomanifold is provided by a planar graph in which all faces including the outer face are triangles. The vertices and the edges of the complex are just the vertices and the edges of the graph. The facets (2-simplices) of the complex are the faces of the planar graph, including the outer face. This is clearly a pure complex, and every edge is contained in exactly two facets. Note that such a graph drawn on a torus or on another 2-manifold works just as well. The *pseudo* part of the definition comes since we are allowing to carry out identifications such as the following: Take a set of vertices that forms an anticlique in the graph and identify all of them to a single vertex. The result is still a two-dimensional psudomanifold. In any event, the uninitiated reader is encouraged to use planar triangulations as a good mental model for a pseudomanifold. Henceforth we shorten pseudomanifold to *psm*.

Let X be a psm on vertex set V. A *k-coloring* of X is a mapping $\varphi : V \to \{1, \ldots, k\}$. A face of X on which φ is $1 : 1$ is said to be *φ-rainbow*. (We only say rainbow when it is clear what coloring is involved.) We are now ready to state and prove a special case of Sperner's lemma that suffices for our purposes.[2]

LEMMA 15. *Let X be an n-dimensional psm. Then for every $(n+1)$-coloring φ of X, the number of φ-rainbow facets of X is even.*

Proof. Consider all pairs $A \supset B$ with A as a facet of X, where B is $(n-1)$-dimensional and φ-rainbow, and $\varphi(B) = [n]$. We count the number of such pairs in two different ways.

Each $(n-1)$-face B with $\varphi(B) = [n]$ participates in exactly two such pairs, once with each of the two facets that contain it. Hence the total count is even.

[2]It is customary to formulate Sperner's lemma as the statement that there is an *odd* number of rainbow facets. However, in those statements the *external* facet has some special status. In the present statement all facets are treated equally. This increases the number of rainbow facets by one.

A facet A contributes to the count iff $\varphi(A) \supseteq [n]$. If $\varphi(A) = [n]$, then there is exactly one element $j \in [n]$ for which $|\varphi^{-1}(j)| = 2$ whereas $|\varphi^{-1}(i)| = 1$ for all $i \neq j$. Consequently, such an A is counted exactly twice. On the other hand, if the facet A is rainbow, then it is counted exactly once.

The claim follows. $\square$

Thus, in particular, if we 3-color the vertices of a triangulated planar graph, so that the outer face is rainbow, then there must be at least one more rainbow face in the triangulated planar graph.

We say that an n-dimensional psm is *colorable* if it has a $(n+1)$-coloring for which no edge is monochromatic. In other words, an $(n+1)$-coloring for which all facets are rainbow.

LEMMA 16. *Let δ be a 2-coloring of a colorable psm X. Then the number of δ-monochromatic facets of X is even.*

Proof. By assumption X is colorable, so let χ be some $(n+1)$-coloring of X in which no edge is monochromatic. Define next a new $(n+1)$-coloring φ via $\varphi := \chi + \delta \mod (n+1)$. By assumption, every facet is χ-rainbow and the addition $(\bmod\,(n+1))$ of a constant value of a monochromatic δ does not change this property. In other words, every δ-monochromatic facet is φ-rainbow. We claim the reverse implication holds as well. Indeed, if δ is not constant on the facet A, then we can find two vertices $x, y \in A$ for which $\delta(x) = 1, \delta(y) = 2$ and $\chi(y) = \chi(x) + 1 \mod (n+1)$. But then no vertex $z \in A$ satisfies $\varphi(z) = \chi(x) + 2 \mod (n+1)$. In other words, a facet is φ-rainbow iff it is δ-monochromatic. By Lemma 15, the proof is complete. $\square$

4.2. MC as a pseudomanifold. Here we prove Theorem 12 by using psm's and Lemma 16. Although psm's can be realized geometrically, we do not refer to such realizations. Still, as mentioned above, planar triangulations can be useful in guiding one's intuition in this area.

Given an MC algorithm in the form of N words, our plan is to construct a psm X that encodes certain possible executions of the MC algorithm. Vertices of X correspond to states of individual players, and facets correspond to reachable configurations. Since we limit ourselves to schedules that involve only n out of the N available players, every facet has n vertices, so that $\dim X = n - 1$.

In the setting of Theorem 12, the scheduler selects one player from each of the $n-2$ pairs (and adds in players P_1, P_2). This gives 2^{n-2} possible initial configurations, which we want to keep as facets. We note, however, that these 2^{n-2} sets do not constitute the collection of facets of a psm. An $(n-2)$-dimensional face that contains one player from each of the $n-2$ pairs and exactly one of P_1, P_2 is contained in exactly one initial facet, in violation of condition (2). Since our intention is to work with psm's we add two auxiliary vertices called A_1 and A_2, where A_1 is viewed as being paired with P_1, and A_2 with P_2. This yields a collection of 2^n *initial facets* that are obtained by making all possible choices of one vertex from each of the n pairs. It is easily verified that this collection constitutes the set of facets of an $(n-1)$-dimensional psm. A facet in this psm is called *auxiliary* or *nonauxiliary* according to whether or not it contains at least one of the auxiliary vertices A_1, A_2. Figure 1 illustrates the situation for $n = 3$. There are six vertices, which correspond to $N = 2n - 2 = 4$ players plus two auxiliary players. The vertices are 3-colored, where each pair of players (say P_1 and A_1) are equally colored. This planar graph has eight faces (including the outer face), which are the $2^3 = 8$ initial facets.

Let us now introduce δ, a 2-coloring of the vertices. We partition the $2n-2$ chairs into two subsets of cardinality $n - 1$ each, called the 0-chairs and the 1-chairs. The

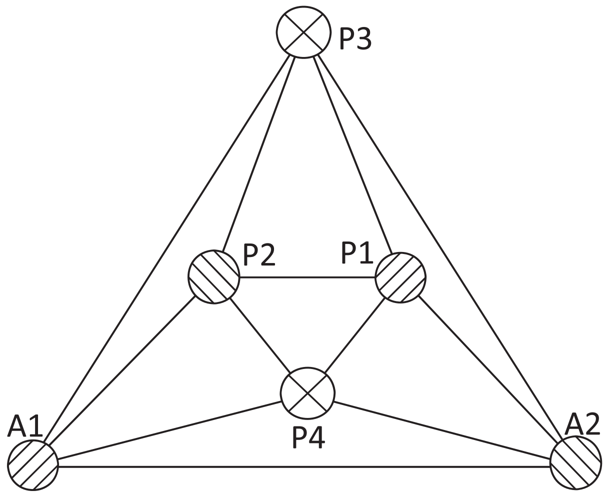

FIG. 1. *A 3-colorable two-dimensional simplicial complex.*

initial chair of P_1, P_2 is a 1-chair whereas the initial chair of A_1, A_2 is a 0-chair.[3] Also, within each pair of players (out of the $n-2$ original pairs), one starts at a 0-chair and the other at a 1-chair, and the δ-color of corresponding vertex in the initial facet is set accordingly to 0 or 1.

PROPOSITION 17. *The collection of all subsets of the* 2^n *initial facets is a colorable* $(n-1)$*-dimensional psm. It has exactly one* δ*-monochromatic auxiliary facet.*

Proof. We already noticed that this collection of sets is indeed a psm. To see that it is colorable, let us associate a unique color to each of the n pairs of vertices. This makes every facet rainbow, as claimed.

In the 2-coloring δ, there is indeed a unique δ-monochromatic initial auxiliary facet. This is the facet that contains A_1, A_2, and the $n-2$ players (one from each pair) who start from a 0-chair. All vertices of this facet have $\delta = 0$. Just for clarity we mention that there is also a nonauxiliary facet all vertices of which have $\delta = 1$ and that includes the vertices P_1 and P_2. $\square$

Starting from the initial system, the rules of MC allow the scheduler to generate new psm's whose facets represent reachable configurations. We remark that unlike the initial configurations, it may happen that several facets correspond to one and the same configuration. This fact will cause no harm to us.

We turn to discuss how a scheduler's move is reflected in pseudomanifold PSM. Consider a configuration in which some players are in conflict and the corresponding facet in PSM. The scheduler may select two players that occupy the same chair, and move either one of them or both. Hence, given the two players and their states (say, corresponding to vertices v_i and v_j in PSM), two new states are exposed by this choice of three possible combined moves. These correspond to two new vertices (say v_i' and v_j') in a new psm. The given configuration can be moved to one of three new configurations, which in our psm representation amounts to splitting each facet σ that contains v_i and v_j into three new facets. That is, each facet $\{v_1, \ldots, v_i, \ldots, v_j, \ldots, v_n\}$

[3] By assumption the cardinality of the set $\{\pi_1[1], \pi_2[1], \pi_3[1], \ldots, \pi_N[1]\}$ is $2n-3$, so that exactly one chair is missing from this set. We associate this missing chair with the auxiliary vertices A_1, A_2. This is, however, just a convenient formality, since there is no player associated with vertices A_1, A_2.

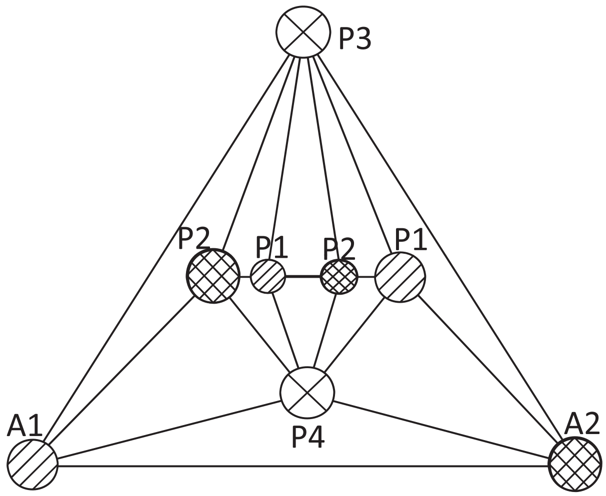

FIG. 2. *The simplicial complex when $n = 3$ after one step by the scheduler.*

is replaced by the three facets $\{v_1, \ldots, v_i, \ldots, v'_j, \ldots, v_n\}$, $\{v_1, \ldots, v'_i, \ldots, v_j, \ldots, v_n\}$, and $\{v_1, \ldots, v'_i, \ldots, v'_j, \ldots, v_n\}$. This completes the description of the new psm PSM'.

We say that the above process *subdivides* the edge $\{v_i, v_j\}$. Figure 2 illustrates the subdivision process when $n = 3$. (It is convenient to have A_1 and A_2 on the outer faces of such drawings, so that edges correspond to straight line segments.)

PROPOSITION 18. *No move of the scheduler can subdivide an auxiliary face.*

Proof. Since vertices A_1 and A_2 do not represent any players, they get never moved by the scheduler, and are, therefore, never involved in a subdivided edge. The rest of the vertices in auxiliary facets are occupying distinct chairs, and hence cannot pair up to create a subdivided edge. ∎

It should be clear how to extend the 2-coloring δ of PSM to PSM'. Assign the two new vertices v'_i and v'_j the 0/1 color of the chairs corresponding to their respective states.

PROPOSITION 19. *The simplicial complex PSM' described above is a colorable pseudomanifold.*

Proof. The colorability of PSM' is inherited from PSM, because every new facet contains the same set of players as its "parent" facet.

To show that PSM' is a psudomanifold we only need to show that each $(n-2)$-face is covered by exactly two facets. To this end we carry out the edge subdivision in two substeps, introducing v'_i and v'_j one at a time. We only need to show that property (2) is maintained under such a substep.

So let w be a new vertex inserted on an edge between u and v in a psm PSM. That is, each facet σ in PSM that contains both u and v is replaced by two facets: $\sigma \setminus \{u\} \cup \{w\}$ and $\sigma \setminus \{v\} \cup \{w\}$. We only need to verify the required property for $(n-2)$-dimensional faces in PSM' that contain the vertex w. Every such an $(n-2)$-face τ contains at most one of the two vertices u, v. If it contains neither of the two, then there are exactly two facets that contain τ, namely, $\tau \cup \{u\}$ and $\tau \cup \{v\}$.

We next turn to the case where v and w are vertices in τ, but u is not. Necessarily $\bar{\tau} = \tau \setminus \{w\} \cup \{u\}$ is an $(n-2)$-face of PSM. As such it is contained in exactly two facets, say, $\bar{\tau} \cup \{x\}$ and $\bar{\tau} \cup \{y\}$. We conclude that $\tau \cup \{x\}$ and $\tau \cup \{y\}$ are the two facets of PSM' that contain τ, as needed. ∎

Proposition 20. *The 2-coloring described for the above simplicial complex PSM' has exactly one monochromatic auxiliary facet.*

Proof. The 2-coloring property is a direct consequence of Propositions 17 and 18. ▯

We are now ready to finish the proof. Consider a psm PSM generated by the process described above, starting from the initial psm and subdividing faces as discussed. By Proposition 19 PSM is colorable. Hence, by Lemma 16, the associated 2-coloring has an even number of monochromatic facets. But Proposition 19 states that exactly one of these facets is auxiliary. Therefore at least one nonauxiliary facet is monochromatic, and this facet represents a reachable configuration θ. But there are exactly $n-1$ 0-chairs and $n-1$ 1-chairs. Therefore there must be two players sharing the same chair in θ, and this allows the scheduler to subdivide the respective edge. Consequently the scheduler can continue to subdivide the psm indefinitely, and this translates to an infinite schedule, as claimed.

5. Oblivious MC algorithms via the probabilistic method.

We start with an observation that puts Theorems 2 and 3 (as well as Theorem 1) in an interesting perspective. The expected number of pairwise conflicts in a random configuration is exactly $\binom{n}{2}/m$. In particular, when $m \gg n^2$, most configurations are *safe* (namely, have no conflicts). Therefore it is not surprising that in this range of parameters n random words would yield an oblivious $MC(n,m)$ algorithm. However, when $m = O(n)$, only an exponentially small fraction of configurations are safe, and the existence of oblivious $MC(n,m)$ algorithms is far from obvious.

5.1. Full words with $O(n)$ chairs, allowing repetitions.

Theorem 2 can be viewed as a (nonconstructive) derandomization of the randomized MC algorithm in which players choose their next chair at random (and future random decisions of players are not accessible to the scheduler). Standard techniques for derandomizing random processes involve taking a union bound over all possible bad events, which in our case corresponds to a union bound over all possible schedules. The immediate scheduler has too many options (and so does the pairwise immediate scheduler), making it infeasible to apply a union bound. For this reason, we shall consider in this section the canonical scheduler, which is just as powerful (see section 2). In every unsafe configuration, the canonical scheduler has just three possible moves to choose from. This allows us to use a union bound. We now prove Theorem 2.

Theorem 2. *For every $N \geq n$, almost every choice of N words of length $cn \log N$ in an alphabet of $m = 7n$ letters is an $MC(n,m)$ winning system with N full words. Moreover, every game on these words terminates in $O(n \log N)$ steps. Here c is an absolute constant.*

Proof. Each of the N words is chosen independently at random as a sequence of L chairs, where each chair in the sequence is chosen independently at random. We show that with high probability (probability tending to 1 as the constant c grows), this choice satisfies Theorem 2.

It is easy to verify that in this random construction, with high probability, all words are full. To see this, note that the probability that chair j is missing from such a random word is $((m-1)/m)^L$. Consequently, the probability that a word chosen this way is not full is $\leq m((m-1)/m)^L$. Therefore, the expected number of nonfull words is $\leq m \cdot N \cdot ((m-1)/m)^L$. But with our choice of parameters $m = 7n$ and $L = cn \log N$, we see that $m \cdot N \cdot ((m-1)/m)^L = o(1)$, provided that c is large enough.

In our approach to the proof we keep track of all possible schedules. To this end we use "a logbook" that is the complete ternary tree $\mathcal{T}$ of depth L rooted at

r. Associated with every node v of $\mathcal{T}$ is a random variable X_v. The values taken by X_v are system configurations. For a given choice of words and an initial system configuration we define the value of X_r to be the chosen initial configuration. Every node v has three children corresponding to the three possible next configurations that are available to the canonical scheduler at configuration X_v.

An important ingredient of the proof is a *potential* function (defined below) that maps system configurations to the nonnegative reals. It is also convenient to define an (artificial) "empty" configuration of 0 potential. Every safe configuration has potential 1, and every nonempty unsafe configuration has potential > 10. If the node u is a descendant of v and the system configuration X_v is safe, then we define X_u to be the empty configuration.

We thus also associate with every node of $\mathcal{T}$ a nonnegative random variable $P = P_v$ that is the potential of the (random) configuration X_v. The main step of the proof is to show that if v_1, v_2, v_3 are the three children of v, then $\sum_{i=1}^{3} \mathbb{E}(P_{v_i}) \leq r\mathbb{E}(P_v)$ for some constant $r \leq 0.99$. (Note that this inequality holds as well if X_v is either safe or empty.) This exponential drop implies that

$$\mathbb{E}\left(\sum_{v \text{ is a leaf of } \mathcal{T}} (P_v)\right) = \sum_{v \text{ is a leaf of } \mathcal{T}} \mathbb{E}(P_v) = o(1)$$

provided that L is large enough. This implies that with probability $1 - o(1)$ (over the choice of random words) all leaves of $\mathcal{T}$ correspond to an empty configuration. In other words every schedule terminates in fewer than L steps.

We turn to the details of the proof. A configuration with i occupied chairs is defined to have potential x^{n-i}, where $x > 1$ is a constant to be chosen later. In a nonempty configuration the potential can vary between 1 and x^{n-1}, and it equals 1 iff the configuration is safe.

Consider a configuration of potential x^{n-i} (with $i < n$), where the canonical pair is (α, β). It has three children representing the move of either α or β, or both. Let us denote $\rho = i/m$ and $\rho' = (i-1)/m$. When a single player moves, the number of occupied chairs can stay unchanged, which happens with probability ρ. With probability $1-\rho$, one more chair will be occupied and the potential gets divided by x. Consider next what happens when both players move. Here the possible outcomes (in terms of number of occupied chairs) depend on whether there is an additional player γ currently co-occupying the same chair as α and β. It suffices to perform the analysis in the less favorable case in which there is no such player γ, as this provides an upper bound on the potential also for the case that there is such a player. With probability $(\rho')^2$, both α and β move to occupied chairs and the potential gets multiplied by x. With probability $\rho'(1-\rho') + (1-\rho')\rho = (\rho+\rho')(1-\rho')$, the number of occupied chairs (and hence the potential) does not change. With probability $(1-\rho')(1-\rho)$, the number of occupied chairs grows by one and the potential gets divided by x.

It follows that if v is a node of $\mathcal{T}$ with children v_1, v_2, v_3 and if the configuration X_v is unsafe and nonempty, then $\sum_{i=1}^{3} \mathbb{E}(P_{v_i}) \leq \mathbb{E}(P_v)(2\rho + 2(1-\rho)/x + (\rho')^2 x + (\rho+\rho')(1-\rho') + (1-\rho)(1-\rho')/x)$. Recall that $x > 1$ and $\rho' < \rho < 1$. This implies that the last expression increases if ρ' is replaced by ρ, and thereafter it is maximized when ρ attains its largest possible value $q = (n-1)/m$. We conclude that

$$\sum_{1}^{3} \mathbb{E}(P_{v_i}) \leq \mathbb{E}(P)(2q + 2(1-q)/x + q^2 x + 2q(1-q) + (1-q)^2/x).$$

We can choose $q = 1/7$ and $x = 23/2$ to obtain $\sum_{i=1}^{3} \mathbb{E}(P_{v_i}) \leq r\mathbb{E}(P_v)$ for $r < 0.99$.

This guarantees an exponential decrease in the expected sum of potentials and hence termination, as we now explain.

It follows that for every initial configuration the expected sum of potentials of all leaves at depth L does not exceed x^{n-1} (the largest possible potential) times r^L. On the other hand, if there is at least one leaf v for which the configuration X_v is neither safe nor empty, then the sum of potentials at depth L is at least $x > 1$. Our aim is to show that with high probability (over the choice of N words), all runs have length $< L$ (i) for every choice of n out of the N words, (ii) each selection of an initial configuration, and (iii) every canonical scheduler's strategy. The n words can be chosen in $\binom{N}{n}$ ways. For every n words, there are L^n possible initial configurations. The probability of length-L run from a given configuration is at most $x^{n-1}r^L$, where $x = 23/2$ and $r < 0.99$. Therefore our claim is proved if $\binom{N}{n} \cdot L^n \cdot x^{n-1}r^L \leq o(1)$. This inequality clearly holds if we let $L = cn \log N$ with c a sufficiently large constant. This completes the proof of Theorem 2. $\square$

A careful analysis of the proof of Theorem 2 shows that it actually works as long as $\frac{m}{n} > 4 + 2\sqrt{2} = 6.828\ldots$. It would be interesting to determine the value of $\liminf_{n\to\infty} \frac{m}{n}$ for which n long enough random words over an m-letter alphabet constitute, with high probability, an oblivious $MC(n,m)$ algorithm.

5.2. Permutations over $O(n)$ chairs. Here we prove Theorem 3.

THEOREM 3. *For every integer $d \geq 1$ there is an $MC(n,m)$ winning system with* $N = n^d$ permutations *on $m = cn$ symbols, where c depends only on d. In fact, this holds for almost every choice of N permutations on $[m]$.*

Our proof of Theorem 2 involves aspects that do not apply in the context of Theorem 3. Theorem 3 deals with random permutations, whereas in the proof of Theorem 2 we use words of length $\Omega(n \log n)$. Recall that the word of a player is a sequence of chairs that the player traverses (upon conflicts) in a cyclic manner. The proof of Theorem 2 establishes that with random words termination is likely in $O(n \log n)$ steps. Making the words longer than the number of steps that suffice for termination avoids the possibility that a player will exhaust his word and wrap-around to the location where he started. If this were to happen, the arguments in the proof of Theorem 2 would not hold anymore, because after a wrap-around occurs future chairs visited are no longer independent of chairs visited in previous steps—they are the same chairs. In Theorem 3 we do not have the option of making words longer than $m = O(n)$, and hence our proof will not be able to avoid dealing with dependencies that arise from wrap-around effects. This is the main source of extra difficulties that need to be dealt with in the proof of Theorem 3 compared to the proof of Theorem 2. These difficulties lead to a substantially different structure of proof for Theorem 3, compared to that of Theorem 2. In particular, our proof of Theorem 3 works with a pairwise immediate scheduler, and unlike the proof of Theorem 2, there does not appear to be any significant benefit (e.g., no significant reduction in the ratio $\frac{m}{n}$) if a canonical scheduler is used instead.

We first prove the special case $N = n$ of Theorem 3, and only later show how to extend the proof to larger values of N.

THEOREM 21. *If $m \geq cn$, where $c > 0$, is a sufficiently large constant, then there is a family of n permutations on $[m]$ which constitute an oblivious $MC(n,m)$ algorithm.*

We actually show that with high probability, a set of random permutations $\pi_1, \ldots, \pi_n$ has the property that in every possible schedule the players visit at most $L = O(m \log m)$ chairs. Our analysis separates between the locations (within the respective words) visited by a schedule, and the contents in these locations (the actual chairs that are placed there by the random permutations). First, when considering a hypothetical schedule, we just view it as a sequence of locations visited. Thereafter, we give random contents to these locations. Finally, we compute the probability that this content is consistent with the assumption that these are the locations visited by the schedule. Consistency requires that the composition of chairs in these locations includes sufficiently many conflicts so as to allow the scheduler to traverse these particular locations. If the probability of consistency is sufficiently low, then Theorem 21 can be proved by using a union bound, as follows. There are m^n possible initial choices of locations in which to start the schedule. (We refer to these here as *initial configurations*.) From each initial configuration we consider all possible sequences of L locations (namely, all possible ways of partitioning L into n nonnegative parts $L_1, \ldots, L_n$ with $\sum L_i = L$, advancing L_i steps on word i). For each such sequence, we fill in the chairs in the locations in the sequence at random and prove that the probability that this sequence represents a possible schedule is extremely small—so small that even if we take a union bound over all initial configurations and over all sequences of length L, we are left with a probability much smaller than 1.

The main difficulty in the proof is that since $L \gg m$, some players may completely traverse their permutation (even more than once) and therefore the chairs in these locations are no longer random. To address this, we partition the sequence of moves into L/t blocks, where in each block players visit a total of t locations. We can and will assume that t divides L. We take $t = \delta m$ for some sufficiently small constant δ, and $n = \epsilon m$, where ϵ is a constant much smaller than δ. This choice of parameters implies that within a block, chairs are essentially random and independent. To deal with dependencies among different blocks, we classify players (and their corresponding permutations) as *light* or *heavy*. A player is light if during the whole schedule (of length L) it visits at most $t/\log m = o(t)$ locations. A player that visits more than $t/\log m$ locations during the whole sequence is heavy. Observe that for light players, the probability of encountering a particular chair in some given location is at most $\frac{1}{m-o(t)} \leq \frac{1+o(1)}{m}$. Hence, the chairs encountered by light players are essentially random and independent (up to negligible error terms). Thus it is the heavy players that introduce dependencies among blocks. Every heavy player visits at least $t/\log m$ locations, so that n_h, the number of heavy players does not exceed $n_h \leq (L \log m)/t = O(\log^2 m)$. The fact that the number of heavy players is small is used in our proof to limit the dependencies among blocks.

The following lemma is used to show that in every block of length t the number of locations that are visited by heavy players is not too large. Consequently, sufficiently many locations are visited by light players. In the lemma, we use the following notation. A segment of k locations in a permutation is said to have *volume* $k - 1$. Given a collection of locations, a chair is *unique* if it appears exactly once in these locations.

LEMMA 22. *Let $n_h \leq m/\log^2 m$ and let $\delta > 0$ be a sufficiently small constant. Consider n random permutations over $[m]$. Select any n_h of the permutations and a starting location in each of them. Choose the next intervals in the selected permutations with total volume t' for some $t/10 \leq t' \leq t$. With probability $1 - o(1)$ for every such set of choices at least $4t'/5$ of the chairs in the chosen intervals are unique.*

Proof. We first note that we will be using the lemma with $n_h = O(\log^2 n)$. Also, if a list of letters contains u unique letters (i.e., they appear exactly once) and r repeated letter (i.e., appearing at least twice), then it has $d = u + r$ distinct letters and length $\lambda \geq u + 2r$. In particular $d \leq (\lambda + u)/2$.

There are $\binom{n}{n_h}$ ways of choosing n_h of the permutations. Then, there are m^{n_h} choices for the initial configuration. We denote by s_i the volume of the ith interval, so that $\sum_{i=1}^{n_h} s_i = t'$. Therefore there are $\binom{t'+n_h-1}{n_h-1} \leq m^{n_h}$ ways of choosing the intervals with total volume t'. Since the volume of every interval is at most t', we have that the probability that a particular chair resides at a particular location in this interval is at most $1/(m-t')$. This is because the permutation is random and at most t' chairs appeared so far in this interval. Therefore the probability that a sequence of t' labels involves less than $0.9t'$ distinct chairs is at most

$$\binom{m}{0.9t'}\left(\frac{0.9t'}{m-t'}\right)^{t'} \leq \left(\frac{em}{0.9t'}\right)^{0.9t'}\left(\frac{0.9t'}{m-t'}\right)^{t'} \leq e^{t'}\left(\frac{m}{m-t'}\right)^{0.9t'}\left(\frac{t'}{m-t'}\right)^{0.1t'}$$
$$\leq 4^{t'}(2\delta)^{0.1t'} \ll e^{-t'}.$$

Explanation. The set of chairs that appear in these intervals can be chosen in $\binom{m}{0.9t'}$ ways. The probability that a particular location in this union of intervals is assigned to a chair from the chosen set does not exceed $\frac{0.9t'}{m-t'}$. In addition $m/(m-t') \leq (1+\delta)$, $t'/(m-t') \leq 2\delta$, and δ is a very small constant.

Now we take a union bound over all choices of n_h permutations, all starting locations and all collection of intervals with total volume t'. It follows that the probability that there is a choice of intervals of volume t' that span $\leq n_h$ permutations and contain fewer than $9t'/10$ distinct chairs is at most

$$m^{3n_h}e^{-t'} = o(1).$$

In the above notation, $\lambda = t'$ and $d \geq 0.9t'$, which yields $u \geq 0.8t'$ as claimed. ∎

Since the conclusion of this lemma holds with probability $1-o(1)$, we can assume that our set of permutations satisfies it. In particular, in every collection of intervals in these permutations with total volume $\frac{t}{10} \leq t' \leq t$ that reside in $O(\log^2 m)$ permutations there are at least $4t'/5$ unique chairs.

As already mentioned, we break the sequence of L locations visited by players into blocks of t locations each. We analyze the possible runs by considering first the *breakpoints profile*, namely, where each block starts and ends on each of the n words. There are m^n possible choices for the starting locations. If, in a particular block, player i visits s_i chairs, then $\sum_{i=1}^{n} s_i = t$. Consequently, the parameters $s_1, \dots, s_n$ can be chosen in $\binom{t+n-1}{n} \leq 2^{t+n}$ ways. There are L/t blocks, so that the total number of possible breakpoints profiles is at most $m^n(2^{t+n})^{L/t} \leq m^n 2^{2L}$. (Here we used the fact that $t > n$.) Clearly, by observing the breakpoints profile we can tell which players are light and which are heavy. We recall that there are at most $O(\log^2 m)$ heavy players and that the premise of Lemma 22 can be assumed to hold.

Let us fix an arbitrary particular breakpoints profile β. We wish to estimate the probability (over the random choice of chairs) that some legal sequence of moves by the pairwise immediate scheduler yields this breakpoints profile β. Let B be an arbitrary block in β. Let $p(B)$ denote the probability over choice of random chairs and *conditioned over contents of all previous blocks in* β that there is a legal sequence of moves by the pairwise immediate scheduler that produces this block B.

LEMMA 23. *For $p(B)$ as defined above we have that $p(B) \leq 8^{-t}$.*

Proof. The total number of chairs encountered in block B is $n \ll t$. (For the initial locations) plus t (for the moves.) Recall that the set of heavy players is determined by the block-sequence β. Hence, within block B it is clear which are the heavy players and which are the light players. Let t_h (resp., $t_\ell = t - t_h$) be the number of chairs visited by heavy (resp., light) players in this block. The proof now breaks into two cases, depending on the value of t_h.

Case 1. $t_h \leq 0.1t$. Light players altogether visit $n + t_\ell$ chairs (n initial locations plus t_ℓ moves). If u of these chair are unique, then they visit at most $(n + t_\ell + u)/2$ distinct chairs. But a chair in this collection that is unique is either (i) one of the n chairs where a player terminates its walk or (ii) a chair that a light player traverses due to a conflict with a heavy player, and there are at most t_h of those. Consequently, the number of distinct chairs visited by light players does not exceed $(n+t_\ell+n+t_h)/2 = t/2 + n$.

Fix the set S of $t/2 + n$ distinct chairs that we are allowed to use. There are $\binom{m}{n+t/2}$ choices for S. Now assign chairs to the locations one by one, in an arbitrary order. Each location has probability of at most $(1 + o(1))\frac{n+t/2}{m}$ of receiving a chair in S. Since we are dealing here with light players, we have exposed only $o(m)$ chairs for each of them (in B and in previous blocks of β), and as mentioned above, this can increase the probability by no more that a $1 + o(1)$ factor.

Hence the probability that the segments traversed by the light players contain only $n + t/2$ chairs is at most

$$\begin{aligned}
&\binom{m}{n+t/2}\left((1+o(1))\frac{n+t/2}{m}\right)^{t_\ell} \\
&\leq \left(\frac{em}{n+t/2}\right)^{n+t/2} 2^{t_\ell}\left(\frac{n+t/2}{m}\right)^{t_\ell} \\
&\leq (2e)^t\left(\frac{n+t/2}{m}\right)^{(t_\ell-t_h)/2-n} \leq (2e)^t\big(t/m\big)^{t/4} < 8^{-t}.
\end{aligned}$$

Here we used that $t_h + t_\ell = t$, $t_h \leq 0.1t$, $t_l \geq 0.9t$, and $n \ll t \ll m$.

Case 2. $t_h \geq 0.1t$. Let us reveal first the chairs visited by the heavy players. By Lemma 22, we find there at least $4t_h/5$ unique chairs. In order that the heavy players traverse these chairs, they must be visited by light players as well. Hence, the t_ℓ locations visited by light players must include all these $0.8t_h$ prespecified chairs. We bound the probability of this as follows. First choose for each of the $0.8t_h$ prespecified chairs a particular location where it should appear in the intervals of light players. The number of such choices is $\leq t_\ell^{0.8t_h}$. As mentioned above, the probability that a particular chair is assigned to some specific location is $(1 + o(1))/m$. Therefore the probability that $0.8t_h$ prespecified chairs appear in the light intervals is at most $t_\ell^{0.8t_h}\big((1+o(1))/m\big)^{0.8t_h}$. Thus the probability that a schedule satisfying the condition of the lemma exists is at most

$$t_\ell^{0.8t_h}\big((1+o(1))/m\big)^{0.8t_h} \leq \big(2t/m\big)^{0.8t_h} \leq \big(2t/m\big)^{t/15} < 8^{-t},$$

where we used that $n \ll t \ll m$. ◻

Lemma 23 implies an upper bound of $p(B)^{L/t} = 8^{-L}$ on the probability there is a legal sequence of moves by the pairwise immediate scheduler that gives rise to breakpoints profile β. Taking a union bound over all block sequences (whose number is at most $m^n 2^{2L} \leq 6^L$, by our choice of $L = Cm \log m$ for a sufficiently large constant C), Theorem 21 is proved.

Observe that the proof of Theorem 21 easily extends to the case that there are $N = m^{O(1)}$ random permutations out of which one chooses n. We simply need to multiply the number of possibilities by N^n, a term that can be absorbed by increasing m, similar to the way the term m^n is absorbed. In Lemma 22 we need to replace $\binom{n}{n_h}$

by $\binom{N}{n_h}$, and the proof goes through without any change (because n_h is so small). This proves Theorem 3.

5.3. Explicit construction with permutations and $m = O(n^2)$. In this section we present for every integer $d \geq 1$ an explicit collection of n^d permutations on $m = O(d^2n^2)$ such that every n of these permutations constitute an oblivious $MC(n, m)$ algorithm. This proves Theorem 4.

We let $LCS(\pi, \sigma)$ stand for the length of the longest common subsequence of the two permutations π and σ, considered cyclically. (That is, we may rotate π and σ arbitrarily to maximize the length of the resulting longest common subsequence). The following easy claim is useful.

PROPOSITION 24. *Let $\pi_1, \ldots, \pi_n$ be permutations of $\{1, \ldots, m\}$ such that $LCS(\pi_i, \pi_j) \leq r$ for all $i \neq j$. If $m > (n-1)r$, then in every schedule none of the π_i is fully traversed.*

Proof. This proof is reached by contradiction. Consider a schedule in which one of the permutations is fully traversed, say, that π_1 is the first permutation to be fully traversed. Each move along π_1 reflects a conflict with some other permutation. Hence, there is a permutation $\pi_i, i > 1$ that has at least $m/(n-1)$ agreements with π_1. Consequently, $r \geq LCS(\pi_1, \pi_i) \geq \frac{m}{(n-1)}$, a contradiction. ∎

This yields an inexplicit oblivious $MC(n, m)$ algorithm with $m = O(n^2)$, since (even exponentially) large families of permutations in $[m]$ exist where every two permutations have an LCS of only $O(\sqrt{m})$. We omit the easy details. On the other hand, we should notice that by [6] this approach is inherently limited and can, at best, yield bounds of the form $m \leq O(n^{3/2})$.

We now present an explicit construction that uses some algebra.

LEMMA 25. *Let p be a prime power, let d be a positive integer, and let $m = p^2$. Then there is an explicit family of $(1 - o(1))m^d$ permutations of an m-element set, where the LCS of every two permutations is at most $4d\sqrt{m}$.*

Proof. Let $\mathbb{F}$ be the finite field of order p. Let $\mathcal{M} := \mathbb{F} \times \mathbb{F}$, and $m = p^2 = |\mathcal{M}|$. Let f be a polynomial of degree $2d$ over $\mathbb{F}$ with vanishing constant term, and let $j \in \mathbb{F}$. We call the set $B_{f,j} = \{(x, f(x) + j) | x \in \mathbb{F}\}$ *a block*. We associate with f the following permutation π_f of $\mathcal{M}$: It starts with an arbitrary ordering of the elements in $B_{f,0}$ followed by $B_{f,1}$ arbitrarily ordered, then of $B_{f,2}$, etc. A polynomial of degree r over a field has at most r roots. It follows that for every two polynomials $f \neq g$ as above and any $i, j \in \mathbb{F}$, the blocks $B_{f,i}$ and $B_{g,j}$ have at most $2d$ elements in common. There are $(p-1) \cdot p^{2d-1} = (1 - o(1))m^d$ such polynomials. There are p blocks in π_f and in π_g, so that $LCS(\pi_f, \pi_g) \leq 4dp$, as claimed. ∎

6. Discussion and open problems. This work originated with the introduction of the concept of oblivious distributed algorithms. In the present paper we concentrated on oblivious MC algorithms, a topic which yields a number of interesting mathematical challenges. We showed that $m \geq 2n - 1$ chairs are necessary and sufficient for the existence of an oblivious MC algorithm with n processors. Still, our construction involves very long words. It is interesting to find explicit constructions with $m = 2n - 1$ chairs and substantially shorter words.

In other ranges of the problem we can show, using the probabilistic method, that oblivious $MC(n, m)$ algorithms exist with $m = O(n)$ and relatively short full words. We still do not have explicit constructions with comparable properties. We would also like to determine $\liminf \frac{m}{n}$ such that n random words over an m letter alphabet typically constitute an oblivious $MC(n, m)$ algorithm.

Computer simulations strongly suggest that for random permutations, a value of $m = 2n-1$ does not suffice. On the other hand, we have constructed (details omitted from this manuscript) oblivious $MC(n, 2n-1)$ algorithms using permutations for $n = 3$ and $n = 4$. (For the latter the proof of correctness is computer-assisted.) For $n \geq 5$ we have neither been able to find such systems (not even in a fairly extensive computer search) nor to rule out their existence.

We do not know how hard it is to recognize whether a given collection of words constitute an oblivious MC algorithm. This can be viewed as the problem of whether some digraph contains a directed cycle or not. The point is that the digraph is presented in a very compact form. It is not hard to place this problem in PSPACE, but is it in a lower complexity class, such as co-NP or P?

REFERENCES

[1] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, and R. Reischuk, *Renaming in an asynchronous environment*, J. ACM, 37 (1990), pp. 524–548.

[2] Y. Afek, H. Attiya, A. Fouren, G. Stupp, and D. Touitou, *Long-Lived Renaming Made Adaptive*, Proceedings of PODC-99, pp. 91–103.

[3] Y. Afek, Y. Babichenko, U. Feige, E. Gafni, N. Linial, and B. Sudakov, *Oblivious collaboration*, Proceedings of the 25th Symposium on Distributed Computing (DISC'11), Lecture Notes in Comput. Sci. 6950, Springer-Verlag, Berlin, 2011, pp. 489–504.

[4] M. Aigner and G. Ziegler, *Proofs from the Book*, 3rd ed., Springer-Verlag, Berlin, 2004.

[5] H. Attiya and S. Rajsbaum, *The combinatorial structure of wait-free solvable tasks*, SIAM J. Comput., 31 (2002), pp. 1286–1313.

[6] P. Beame, E. Blais, and D. Ngoc, *Longest Common Subsequences in Sets of Permutations*, http://arxiv.org/pdf/0904.1615.pdf.

[7] E. Borowsky and E. Gafni, *Generalized FLP impossibility results for t-resilient asynchronous computations*, Proceedings of the 25th ACM Symposium on Theory of Computing (STOC'93), 1993, pp. 91–100.

[8] E. Gafni and S. Rajsbaum, *Musical Benches*, Proceedings of the 19th Internatonal Symposium on Distributed Computing (DISC'05), Lecture Notes in Comput. Sci. 3724, Springer-Verlag, Berlin, 2005, pp. 63–77.

[9] P. Hall, *On representatives of subsets*, J. Lond. Math. Soc., 10 (1935), pp. 26–30.

[10] M.P. Herlihy and N. Shavit, *The topological structure of asynchronous computability*, J. ACM, 46 (1999), pp. 858–923.

[11] M. Saks and F. Zaharoglou, *Wait-free k-set agreement is impossible: The topology of public knowledge*, SIAM J. Comput., 29 (2000), pp. 1449–1483.