The Parallel Complexity of Coloring Games

Guillaume Ducoffe

▶ **To cite this version:**

Guillaume Ducoffe. The Parallel Complexity of Coloring Games. 9th International Symposium, SAGT 2016, Sep 2016, Liverpool, United Kingdom. pp.27-39, 10.1007/978-3-662-53354-3_3 . hal-01361056

HAL Id: hal-01361056
https://hal.archives-ouvertes.fr/hal-01361056

Submitted on 6 Sep 2016

The parallel complexity of coloring games [*]

Guillaume Ducoffe[1]

Université Côte d'Azur, Inria, CNRS, I3S, France

Abstract. We wish to motivate the problem of finding *decentralized lower-bounds* on the complexity of computing a Nash equilibrium in graph games. While the centralized computation of an equilibrium in polynomial time is generally perceived as a positive result, this does not reflect well the reality of some applications where the game serves to implement distributed resource allocation algorithms, or to model the social choices of users with limited memory and computing power. As a case study, we investigate on the parallel complexity of a game-theoretic variation of graph coloring. These "coloring games" were shown to capture key properties of the more general welfare games and Hedonic games. On the positive side, it can be computed a Nash equilibrium in polynomial-time for any such game with a local search algorithm. However, the algorithm is time-consuming and it requires polynomial space. The latter questions the use of coloring games in the modeling of information-propagation in social networks. We prove that the problem of computing a Nash equilibrium in a given coloring game is PTIME-hard, and so, it is unlikely that one can be computed with an efficient distributed algorithm. The latter brings more insights on the complexity of these games.

1 Introduction

In algorithmic game theory, it is often the case that a problem is considered "tractable" when it can be solved in polynomial time, and "difficult" only when it is NP-hard or it is PLS-hard to find a solution. On the other hand, with the growing size of real networks, it has become a boiling topic in (non game-theoretic) algorithmic to study on the finer-grained complexity of polynomial problems [16]. In our opinion, the same should apply to graph games when they serve as a basis for new distributed algorithms. We propose to do so in some cases when it can be easily computed a Nash equilibrium in polynomial time. The following case study will make use of well-established parallel and space complexity classes to better understand the hardness of a given graph game.

Precisely, we investigate on a "coloring game", first introduced in [14] in order to unify classical upper-bounds on the chromatic number. Since then it has been rediscovered many times, attracting attention on the way in the study of information propagation in wireless sensor networks [4] and in social networks [12]. We choose to consider this game since it is a good representative of the *separable welfare games* – proposed in [13] as a game-theoretic toolkit for distributed

resource allocation algorithms – and the *additively separable symmetric Hedonic games* [3]. A coloring game is played on an undirected graph with each vertex being an agent (formal definitions will be given in the technical sections of the paper). Agents must choose a colour in order to construct a proper coloring of the graph. The individual goal of each agent is to maximize the number of agents with the same colour as hers. Furthermore, it can always be computed a Nash equilibrium in polynomial time with a simple local-search algorithm [6, 12, 14].

However, for n-vertex m-edge graphs, the above-mentioned algorithm has $\mathcal{O}(m + n\sqrt{n})$-time complexity and $\mathcal{O}(n+m)$-space complexity. Therefore, when the graph gets larger, potential applications of coloring games as a *computational mechanism design* (*e.g.*, in order to assign frequencies in sensor networks in a distributed fashion, or to model the behaviour of social network users with limited power and storage) can be questioned. In particular, the authors in [11] report on the limited abilities of human subject networks to solve a coloring problem. In this note, we will investigate on the belonging of our problem – the computation of a Nash equilibrium in coloring games – to some complexity classes that are related to parallel and space complexity. Our goal in doing so is to bring more insights on the complexity of the problem.

Related work. Apart from lower-bounds in communication complexity [7], we are not aware of any analysis of decentralized complexity in game theory. Closest to our work are the studies on the sequential complexity of Hedonic games. Deciding whether a given Hedonic game admits a Nash equilibrium is NP-complete [1]. Every additively separable symmetric Hedonic games has a Nash equilibrium but it is PLS-complete to compute one [8]. Coloring games are a strict subclass where the local-search algorithm terminates on a Nash equilibrium within a polynomial number of steps. We will go one step further by considering their parallel complexity, something we think we are the first to study.

In [4], they introduced a distributed algorithm in order to compute the Nash equilibrium of a given coloring game. Their algorithm is a natural variation of the classical local-search algorithm for the problem, however, it does not speed up the computation of equilibria (at least theoretically). In addition, each agent needs to store locally the colouring of the graph at any given step, that implies *quadratic* space and communication complexity. Additional related work is [6, 12], where it is studied the number of steps of more elaborate local-search algorithms when up to k players are allowed to *collude* at each step. Informally, collusion means that the players can simultaneously change their colours for the same new colour provided they all benefit from the process (note that the classical local-search algorithm corresponds to the case $k = 1$).

Contributions. We prove that the problem of computing a Nash equilibrium in a given coloring game is PTIME-hard (Theorem 2). This is hint that the problem is inherently *sequential*, *i.e.*, it is unlikely the computation of an equilibrium can be sped up significantly on a parallel machine with polynomially many processors. In particular, our negative result applies to the *distributed* setting since any distributed algorithm on graphs can be simulated on a parallel machine with

one processor per edge and per vertex. By a well-known relationship between space and parallel complexity [15], Theorem 2 also extends to show that no *space efficient* algorithm for the problem (say, within logarithmic workspace) can exist. Altogether, this may be hint that coloring games are a too powerful computational mechanism design for "lightweight" distributed applications.

Our reduction is from the standard MONOTONE CIRCUIT VALUE problem. However, the gadgets needed are technically challenging, and we will need to leverage nontrivial properties of coloring games in order to prove its correctness. Definitions and useful background will be given in Section 2. We will detail our reduction in Section 3 before concluding this paper in Section 4.

2 Definitions and notations

We use the graph terminology from [2]. Graphs in this study are finite, simple, and unweighted.

Coloring games. Let $G = (V, E)$ be a graph. A *coloring* of G assigns a positive integer, taken in the range $\{1, \ldots, n\}$, to each of the n vertices in V. For every i, let L_i be the subset of vertices coloured i. We name L_i a *colour class* in what follows. Nonempty colour classes partition the vertex set V. The partition is a *proper coloring* when no two adjacent vertices are assigned the same colour, *i.e.*, for every $1 \leq i \leq n$ and for every $u, v \in L_i$, $\{u, v\} \notin E$.
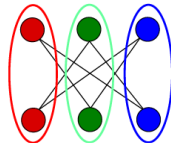

Fig. 1. Proper coloring of a graph G. Each colour class is represented by an ellipse. Every agent receives unit payoff.

Every graph G defines a *coloring game* whose n agents are the vertices in V. The strategy of an agent is her colour. Furthermore, every $v \in L_i$ receives payoff: -1 if there is $u \in L_i$ s.t. $\{u, v\} \in E$ (in which case, the coloring is not proper), and $|L_i| - 1$ otherwise. We refer to Figure 1 for an illustration. Finally, a *Nash equilibrium* of the coloring game is any coloring of G where no agent can increase her payoff by changing her strategy. In particular, the proper coloring in Figure 1 is a Nash equilibrium. More generally, observe that a Nash equilibrium in this game is always a proper coloring of G. In what follows, we will focus on the computation of Nash equilibria in coloring games.

Theorem 1 ([6, 12]). *For any coloring game that is specified by an n-vertex m-edge graph $G = (V, E)$, a Nash equilibrium can be computed in $\mathcal{O}(m + n\sqrt{n})$-time and $\mathcal{O}(n + m)$-space.*

Parallel complexity. Computations are performed on a parallel random-access machine (PRAM, see [9]) with an unlimited amount of processors. However, as

stated in the conclusion, our results also apply to more realistic parallel complexity classes. In what follows, we will use the fact that processors are numbered. We will handle with read/write conflicts between processors with the strategy CREW-PRAM (concurrent read, exclusive write). Let PTIME contain the decision problems that can be solved in *sequential* polynomial-time (that is, with a single processor). Problem A reduces to problem B if given an oracle to solve B, A can be solved in polylogarithmic-time with a polynomial number of processors. In particular, a problem B is PTIME-hard if every problem in PTIME reduces to B (this is formally defined as quasi-PTIME-hardness in [9]). Such reductions are finer-grained than the more standard logspace reductions.

3   Main Result

Theorem 2. *Computing a Nash equilibrium for coloring games is PTIME-hard.*

In order to prove Theorem 2, we will reduce from a variation of the well-known MONOTONE CIRCUIT VALUE problem, defined as follows.

---

*Problem 1 (*MONOTONE CIRCUIT VALUE*).*

Input: A boolean circuit $\mathcal{C}$ with m gates and n entries, a word $w \in \{0,1\}^n$ such that:
  − the gates are either AND-gates or OR-gates;
  − every gate has exactly two entries (in-degree two);
  − a *topological ordering* of the gates is given, with the mth gate being the output gate.
Question: Does $\mathcal{C}$ output 1 when it takes w as input ?

MONOTONE CIRCUIT VALUE is proved to be PTIME-complete in [9].

3.1   The reduction

Let $\langle \mathcal{C}, w \rangle$ be any instance of MONOTONE CIRCUIT VALUE. We will reduce it to a coloring game as follows. Let $\mathcal{G} := (g_1, g_2, \ldots, g_m)$ be the gates of the circuit, that are topologically ordered.

Construction of the gate-gadgets. For every $1 \leq j \leq m$, the jth gate will be simulated by a subgraph $G_j = (V_j, E_j)$ with $12(n+j) - 9$ vertices. We refer to Figure 2 for an illustration. Let us give some intuition for the following construction of G_j. We aim at simulating the computation of the (binary) output of all the gates in $\mathcal{C}$ when it takes w as input. To do that, given a supergraph G of G_j (to be defined later), and a fixed Nash equilibrium for the coloring game that is defined on G, we aim at guessing the output of the jth gate from the subcoloring of G_j. More precisely, the subcoloring will encode a "local certificate" that indicates which values on the two entries of g_j cause the output.

Observe that to certify that an OR-gate outputs 1, it suffices to show that it receives 1 on any one of its two entries, whereas for an AND-gate it requires to

show that it outputs 1 on its two entries. Since by de Morgan's laws, the negation of an AND-gate can be transformed into an OR-gate and vice-versa, therefore, we need to distinguish between three cases in order to certify the output of the gate. So, the vertices in V_j are partitioned in three subsets of equal size $4(n + j) - 3$, denoted by V_j^1, V_j^2, V_j^3. Furthermore, for every $1 \le t \le 3$, every vertex in V_j^t is adjacent to every vertex in $V_j \setminus V_j^t$.
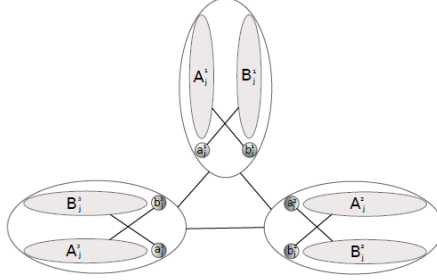

Fig. 2. Gadget subgraph G_j representing the jth gate. An edge between two subsets of vertices (delimited by an ellipse) denotes the existence of a complete bipartite subgraph.

Let us now describe the structure of the three (isomorphic) subgraphs $G_j[V_j^t] = (V_j^t, E_j^t)$ with $1 \le t \le 3$. Informally, we will need this internal structure in order to ensure that every of the three subsets V_j^t will behave as a "truthful" certificate to decide on the output of the gate; *i.e.*, only a few vertices of V_j will be used to certify the output of the jth gate, while all others will be divided into artificial aggregates that we name "private groups" whose role is to ensure "truthfulness" of the certificate (this will be made clearer in the following). There are two nonadjacent vertices $a_j^t, b_j^t \in V_j^t$ playing a special role. The other vertices in $V_j^t \setminus \{a_j^t, b_j^t\}$ are partitioned in two subsets A_j^t, B_j^t of respective size $2(n + j) - 3$ and $2(n + j) - 2$. The sets A_j^t, B_j^t are called the *private groups* of a_j^t, b_j^t. Furthermore, every vertex in A_j^t is adjacent to every vertex in $V_j^t \setminus (A_j^t \cup \{a_j^t\})$, similarly every vertex in B_j^t is adjacent to every vertex in $V_j^t \setminus (B_j^t \cup \{b_j^t\})$.

Since all edges are defined above independently the one from the other, the graph $G_j[V_j^1] = (V_j^1, E_j^1)$ (encoded by its adjacency lists) can be constructed with $|V_j^1| + |E_j^1| = 4(n + j)^2 - 2(n + j) - 2$ processors simply by assigning the construction of each vertex and each edge to a different processor. Note that each processor can decide on the vertex, resp. the edge, it needs to compute from its number. Overall, it takes $\mathcal{O}(\log(n + j))$-time in order to construct $G_j[V_j^1]$ in parallel. The latter can be easily generalized in order to construct G_j in $\mathcal{O}(\log(n + j))$-time with $|V_j| + |E_j|$ processors. Therefore, the graphs $G_1, G_2, \ldots, G_m$ can be constructed in parallel in $\mathcal{O}(\log(n + m))$-time with $\sum_{j=1}^{m}(|V_j| + |E_j|)$ processors, that is polynomial in $n + m$.

Construction of the graph. Let $X = \{x_1, x_1', \ldots, x_i, x_i', \ldots, x_n, x_n'\}$ contain $2n$ nonadjacent vertices, that are two vertices per letter in the binary word w. The graph $G = (V, E)$ for the reduction has vertex-set $V = X \cup \left(\bigcup_{j=1}^{m} V_j \right)$. In particular, it has $2n - 9m + 6m(m + 2n + 1)$ vertices. Furthermore, $G[V_j]$ is

isomorphic to G_j for every $1 \le j \le m$. In order to complete our reduction, let us now describe how our gadgets are connected the one with the other.

For technical reasons, we will need to make adjacent every vertex in the private group A_j^t (resp. B_j^t), with $1 \le j \le m$ and $1 \le t \le 3$, to every vertex in $V \setminus V_j$. By doing so, note that every vertex in $V \setminus (A_j^t \cup \{a_j^t\})$ is adjacent to every vertex in A_j^t (resp., every vertex in $V \setminus (B_j^t \cup \{b_j^t\})$ is adjacent to every vertex in B_j^t). Furthermore, each edge is defined independently the one from the other. Hence, similarly as above, $\sum_{j=1}^{m} \sum_{t=1}^{3} (|A_j^t| + |B_j^t|)|V \setminus V_j|$ processors are sufficient in order to construct these edges in $\mathcal{O}(\log(n+m))$-time, that is polynomial in $n + m$.
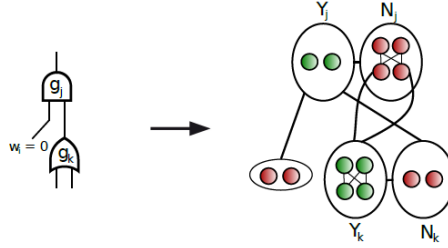

Fig. 3. Edges in G to simulate the two connections of an AND-gate in the circuit.

Finally, we recall that for every j, there are three cases to distinguish in order to decide on the output of the jth gate, with each case being represented with some subset V_j^t. The union of subsets representing a *positive* certificate (output 1) is named Y_j, while the union of those representing a *negative* certificate (output 0) is named N_j. In particular, if the jth gate is an OR-gate, let $Y_j := \{a_j^1, b_j^1, a_j^2, b_j^2\}$ and $N_j := \{a_j^3, b_j^3\}$ (it suffices to receive 1 on one input). Else, the jth gate is an AND-gate, so, let $Y_j := \{a_j^1, b_j^1\}$ and $N_j := \{a_j^2, b_j^2, a_j^3, b_j^3\}$.

Suppose the jth gate is an OR-gate (the case when it is an AND-gate follows by symmetry, up to interverting Y_j with N_j, see also Figure 3). Let us consider the first entry of the gate. There are two cases. Suppose that it is the ith entry of the circuit, for some $1 \le i \le n$. If $w_i = 0$ then we make both x_i, x_i' adjacent to both a_j^1, b_j^1; else, $w_i = 1$, we make both x_i, x_i' adjacent to both a_j^3, b_j^3. Else, the entry is some other gate of the circuit, and so, since gates are topologically ordered, it is the kth gate for some $k < j$. We make every vertex in N_k adjacent to both a_j^1, b_j^1, and we make every vertex in Y_k adjacent to both a_j^3, b_j^3.

The second entry of the gate is similarly considered, up to replacing above the two vertices a_j^1, b_j^1 with a_j^2, b_j^2. We refer to Figure 3 for an illustration. In particular, observe that there is only a constant number of edges that are added at this step for each gate. Furthermore, the construction of these new edges only requires to read the two in-neighbours of the gate in the circuit $\mathcal{C}$. As a result, the last step can be done in parallel in $\mathcal{O}(\log(n+m))$-time with m processors.

3.2 Structure of a Nash equilibrium

The graph $G = (V, E)$ of our reduction (constructed in Section 3.1) defines a coloring game. Let us fix any Nash equilibrium for this game (that exists by

Theorem 1). We will show that it is sufficient to know the colour of every vertex in $Y_m \cup N_m$ in order to decide on the output of the circuit $\mathcal{C}$ (recall that the mth gate is the output gate). To prove it, we will need the following technical claims in order to gain more insights on the structure of the equilibrium.
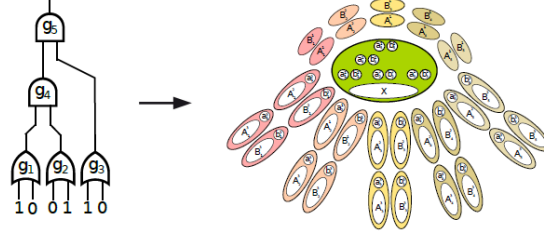

Fig. 4. A boolean circuit (left) with a Nash equilibrium of the coloring game from our reduction (right). Each colour class is represented with an ellipse. Intuitively, vertices in the central colour class simulate the computation of the output. Other colour classes contain a private group and they are "inactive".

More precisely, we will prove that there are exactly $6m + 1$ colour classes, that are one colour class per private group A_j^t or B_j^t and an additional colour for the vertices in X. The intuition is that there are $2(n + m)$ vertices in one special colour class (including X) that simulates the computation of the output of $\mathcal{C}$, whereas all other vertices are "trapped" with the vertices in their respective private group. We refer to Figure 4 for an illustration.

Claim 1. *For every j, any colour class does not contain more than two vertices in every $Y_j \cup N_j$. Furthermore, if it contains exactly two vertices in $Y_j \cup N_j$ then these are a_j^t, b_j^t for some $1 \le t \le 3$.*

Proof. A Nash equilibrium is a proper coloring of G. Therefore, since any two vertices in different subsets among V_j^1, V_j^2, V_j^3 are adjacent by construction, they cannot have the same colour. Since $Y_j \cup N_j = \{a_j^1, b_j^1, a_j^2, b_j^2, a_j^3, b_j^3\}$ and $a_j^t, b_j^t \in V_j^t$ for every $1 \le t \le 3$, the claim follows directly. $\diamond$

Claim 2. *Any two vertices that are in a same private group have the same colour. Similarly, x_i and x_i' have the same colour for every $1 \le i \le n$.*

Proof. Let S be either a private group ($S = A_j^t$ or $S = B_j^t$ for some $1 \le j \le m$ and $1 \le t \le 3$), or a pair representing the same letter of word w (i.e., $S = \{x_i, x_i'\}$ for some $1 \le i \le n$). Let $v \in S$ maximize her payoff and let c be her colour. Note that v receives payoff $|L_c| - 1$ with L_c being the colour class composed of all the vertices with colour c. Furthermore, every $u \in S$ receives payoff lower than or equal to $|L_c| - 1$ by the choice of v. In such case, every $u \in S$ must be coloured c, or else, since the adjacency and the nonadjacency relations are the same for u and v (they are twins), furthermore u, v are nonadjacent, the agent u would increase her payoff to $|L_c|$ by choosing c as her new colour, thus contradicting the hypothesis that we are in a Nash equilibrium. $\diamond$

The argument we use in Claim 2 is that twin vertices must have the same colour. In what follows, we will use the same argument under different disguises.

Claim 3. *Let $1 \leq j \leq m$ and $1 \leq t \leq 3$. Either A_j^t or $A_j^t \cup \{a_j^t\}$ is a colour class, and in the same way either B_j^t or $B_j^t \cup \{b_j^t\}$ is a colour class. Furthermore, either $B_j^t \cup \{b_j^t\}$ is a colour class, or a_j^t and b_j^t have the same colour.*

Proof. Recall that a Nash equilibrium is a proper coloring of G. Since a_j^t is the only vertex in $V \setminus A_j^t$ that is nonadjacent to A_j^t, furthermore every two vertices in A_j^t have the same colour by Claim 2, therefore, either A_j^t or $A_j^t \cup \{a_j^t\}$ is a colour class. Similarly, either B_j^t or $B_j^t \cup \{b_j^t\}$ is a colour class. In particular, suppose that b_j^t does not have the same colour as her private group. Then, she must receive payoff at least $|B_j^t| = 2(n + j) - 2$ (else, she would increase her payoff by choosing the same colour as her private group, thus contradicting the hypothesis that we are in a Nash equilibrium). Furthermore, there can be only vertices in $V \setminus (A_j^t \cup B_j^t)$ with the same colour c as b_j^t. Suppose for the sake of contradiction that a_j^t does not have colour c. There are two cases to be considered.

Suppose that $A_j^t \cup \{a_j^t\}$ is a colour class. Then, a_j^t receives payoff $|A_j^t| = 2(n + j) - 3$. In such case, since a_j^t and b_j^t are twin vertices in $G \setminus (A_j^t \cup B_j^t)$, vertex a_j^t could increase her payoff to at least $2(n + j) - 1$ by choosing c as her new colour, thus contradicting the hypothesis that we are in a Nash equilibrium.

Else, a_j^t and b_j^t do not have the same colours as their respective private groups. In such case, A_j^t and B_j^t are colour classes, hence we can constrain ourselves to the subgraph $G \setminus (A_j^t \cup B_j^t)$. In particular, the constriction of the Nash equilibrium to the subgraph must be a Nash equilibrium of the coloring game defined on $G \setminus (A_j^t \cup B_j^t)$. Since a_j^t and b_j^t are twin vertices in $G \setminus (A_j^t \cup B_j^t)$, they must have the same colour by a similar argument as for Claim 2.

As a result, a_j^t must have colour c in both cases, that proves the claim. $\diamond$

We recall that we aim at simulating the computation of the output of all the gates in $\mathcal{C}$. To do that, we will prove the existence of a special colour class containing X and some pair in $Y_j \cup N_j$ for every j. Intuitively, the two vertices of $Y_j \cup N_j$ are used to certify the output of the jth gate. However, this certificate is "local" in the sense that it assumes the output of the $j - 1$ smaller gates to be already certified. Therefore, we need to prove that there can be no "missing gate", *i.e.*, every gate is represented in the special colour class.

Claim 4. *Let c be a colour such that $L_c \not\subseteq X$ and L_c does not intersect any private group (A_j^t or B_j^t for any $1 \leq j \leq m$ and $1 \leq t \leq 3$).*

Then, $X \subseteq L_c$ and there exists an index j_0 such that the following holds true: $|L_c \cap (Y_j \cup N_j)| = 2$ for every $1 \leq j \leq j_0$, and $L_c \cap (Y_j \cup N_j) = \emptyset$ for every $j_0 + 1 \leq j \leq m$.

Proof. By the hypothesis $L_c \not\subseteq X$ and L_c does not intersect any private group, so, there is at least one vertex of $\bigcup_{j=1}^m (Y_j \cup N_j)$ with colour c. Let j_0 be the largest index j such that there is a vertex in $Y_j \cup N_j$ with colour c. Since by Claim 1, there can be no more than two vertices of $Y_j \cup N_j$ that are in L_c for every j, therefore, by maximality of j_0 we get $|L_c| \leq |X| + 2j_0 = 2(n + j_0)$. In particular, observe that if $|L_c| = 2(n + j_0)$ then $X \subseteq L_c$ and for every $1 \leq j \leq j_0$ there are exactly two vertices in $Y_j \cup N_j$ with colour c. So, let us prove that $|L_c| = 2(n + j_0)$,

that will prove the claim. By the choice of j_0, there is some $1 \leq t \leq 3$ such that $a_{j_0}^t \in L_c$ or $b_{j_0}^t \in L_c$. In particular, $|L_c| \geq \min\{|A_{j_0}^t|, |B_{j_0}^t|\} + 1 = 2(n + j_0) - 2$ or else, every vertex $v_{j_0}^t \in L_c \cap \{a_{j_0}^t, b_{j_0}^t\}$ would increase her payoff by choosing the colour of the vertices in her private group (that is a colour class by Claim 3), thus contradicting the hypothesis that we are in a Nash equilibrium.

We prove as an intermediate subclaim that for any $1 \leq j \leq j_0 - 1$ such that $L_c \cap (Y_j \cup N_j) \neq \emptyset$, there is some $1 \leq t' \leq 3$ such that $a_j^{t'}, b_j^{t'} \in L_c$. Indeed, in this situation, there is some t' such that $a_j^{t'} \in L_c$ or $b_j^{t'} \in L_c$. If $b_j^{t'} \in L_c$ then we are done as by Claim 3, $a_j^{t'} \in L_c$. Otherwise, $b_j^{t'} \notin L_c$ and we prove this case cannot happen. First observe that $a_j^{t'} \in L_c$ in this case. Furthermore, since $a_j^{t'}$ and $b_j^{t'}$ do not have the same colour we have by Claim 3 that $B_j^{t'} \cup \{b_j^{t'}\}$ is a colour class. In this situation, $b_j^{t'}$ receives payoff $2(n + j) - 2 \leq 2(n + j_0 - 1) - 2 < |L_c|$. Since in addition $a_j^{t'}$ and $b_j^{t'}$ are twins in $G \setminus (A_j^{t'} \cup B_j^{t'})$, vertex $b_j^{t'}$ could increase her payoff by choosing colour c, thus contradicting that we are in a Nash equilibrium. This proves $a_j^{t'}, b_j^{t'} \in L_c$, and so, the subclaim.

By the subclaim, there is an even number $2k$ of vertices in $\bigcup_{j=1}^{j_0-1}(Y_j \cup N_j)$ with colour c, for some $k \leq j_0 - 1$. Similarly, since by Claim 2 the vertices x_i, x_i' have the same colour for every $1 \leq i \leq n$, $|X \cap L_c| = 2n'$ for some $n' \leq n$. Now there are two cases to be considered.

Suppose that $b_{j_0}^t \in L_c$. Then, by Claim 3 $a_{j_0}^t \in L_c$. Furthermore $|L_c| \geq 2(n + j_0) - 1$ or else, vertex $b_{j_0}^t$ would increase her payoff by choosing the colour of the vertices in $B_{j_0}^t$ (that is a colour class by Claim 3), thus contradicting the hypothesis that we are in a Nash equilibrium. As a result, $|L_c| = 2(n' + k + 1) \geq 2(n + j_0) - 1$, that implies $n' + k \geq n + j_0 - 1$, and so, $|L_c| \geq 2(n + j_0)$, as desired.

Else, $b_{j_0}^t \notin L_c$ and we prove this case cannot happen. First observe that $a_{j_0}^t \in L_c$. Furthermore, $|L_c| = 2(n' + k) + 1 \geq 2(n + j_0) - 2$, that implies $n' + k \geq n + j_0 - 1$, and so, $|L_c| \geq 2(n + j_0) - 1$. However, since $a_{j_0}^t$ and $b_{j_0}^t$ do not have the same colour, $B_{j_0}^t \cup \{b_{j_0}^t\}$ is a colour class by Claim 3. In particular, $b_{j_0}^t$ receives payoff $2(n + j_0) - 2 < |L_c|$. Since $a_{j_0}^t, b_{j_0}^t$ are twins in $G \setminus (A_{j_0}^t \cup B_{j_0}^t)$, vertex $b_{j_0}^t$ could increase her payoff by choosing colour c, thus contradicting that we are in a Nash equilibrium.

Altogether, $|L_c| \geq 2(n + j_0)$, that proves the claim. $\diamond$

We point out that by combining Claim 1 with Claim 4, one obtains that for every $1 \leq j \leq m$, there are either zero or two vertices in $Y_j \cup N_j$ in each colour class not containing a private group, and in case there are two vertices then these are a_j^t, b_j^t for some $1 \leq t \leq 3$.

Claim 5. *Any two vertices in X have the same colour. Furthermore, for every $1 \leq j \leq m$, every vertex in $Y_j \cup N_j$ either has the same colour as vertices in X or as vertices in her private group.*

Proof. Let L_c be any colour class with at least one vertex in $\bigcup_{j=1}^m (Y_j \cup N_j)$. Let j_0 be the largest index j such that there is a vertex in $Y_j \cup N_j$ with colour c. In order to prove the claim, there are two cases to be considered. Suppose that

$L_c \neq A^t_{j_0} \cup \{a^t_{j_0}\}$ and $L_c \neq B^t_{j_0} \cup \{b^t_{j_0}\}$ for any $1 \leq t \leq 3$. We will prove that $X \subseteq L_c$, that will imply that L_c is unique in such a case, and so, will prove the claim. By the choice of colour c, $L_c \not\subseteq X$. Further, observe that there can be no private group with a vertex in L_c. As a result, this case follows directly from Claim 4.

Else, either $L_c = A^t_{j_0} \cup \{a^t_{j_0}\}$ or $L_c = B^t_{j_0} \cup \{b^t_{j_0}\}$ for some $1 \leq t \leq 3$, and we may assume that it is the case for any colour class L_c that contains at least one vertex in $\bigcup_{j=1}^m (Y_j \cup N_j)$ (or else, we are back to the previous case). So, let us constrain ourselves to the subgraph $G[X]$. In particular, the constriction of the Nash equilibrium to the subgraph must be a Nash equilibrium of the coloring game defined on $G[X]$. Since the vertices in X are pairwise nonadjacent, they must form a unique colour class in such case, that proves the claim. $\diamond$

We will need a "truthfulness" property to prove correctness of our reduction. Namely, the value of the output of any gate in the circuit must be correctly guessed from the agents with the same colour as vertices in X.

Claim 6. *Let $1 \leq j_0 \leq m$ such that for every $1 \leq j \leq j_0$, there is at least one vertex in $Y_j \cup N_j$ with the same colour c_0 as all vertices in X. Then for every $1 \leq j \leq j_0$, $L_{c_0} \cap Y_j \neq \emptyset$ if and only if the output of the jth gate is 1.*

Proof. In order to prove the claim by contradiction, let $1 \leq j_1 \leq j_0$ be the smallest index j such that either $Y_j \cap L_{c_0} = \emptyset$ and the output of the jth gate is 1 (false negative) or $Y_j \cap L_{c_0} \neq \emptyset$ and the output of the jth gate is 0 (false positive). We will show that in such case, there is an edge with two endpoints of colour c_0, hence the coloring is not proper, thus contradicting the hypothesis that we are in a Nash equilibrium. Note that since by de Morgan's laws, the negation of an AND-gate can be transformed into an OR-gate and vice-versa, both cases are symmetrical, and so, we can assume w.l.o.g. that the j_1^{th} gate is an OR-gate. There are two subcases to be considered.

Suppose that the output of the j_1^{th} gate is 0 (false positive). In such case, $Y_{j_1} \cap L_{c_0} \neq \emptyset$. Let us consider the first entry of the gate. If it is the ith entry of the circuit for some $1 \leq i \leq n$ then $w_i = 0$ (because the output of the j_1^{th} gate is 0) and so, by construction, $x_i, x'_i \in L_{c_0}$ are adjacent to $a^1_{j_1}, b^1_{j_1}$. Else, it is the kth gate of the circuit for some $k < j_1$. By minimality of j_1, since the output of the kth gate must be 0 (because the output of the j_1^{th} gate is 0), $Y_k \cap L_{c_0} = \emptyset$, and so, $N_k \cap L_{c_0} \neq \emptyset$. By construction, every vertex in N_k is adjacent to $a^1_{j_1}, b^1_{j_1}$. As a result, $a^1_{j_1}, b^1_{j_1}$ have a neighbour in L_{c_0} in this subcase. We can prove similarly (by considering the second entry of the gate) that $a^2_{j_1}, b^2_{j_1}$ have a neighbour in L_{c_0} in this subcase. The latter implies the existence of an edge with both endpoints in L_{c_0} since $Y_{j_1} = \{a^1_{j_1}, b^1_{j_1}, a^2_{j_1}, b^2_{j_1}\}$.

Else, the output of the j_1^{th} gate is 1 (false negative). In such case, $Y_{j_1} \cap L_{c_0} = \emptyset$, hence $N_{j_1} \cap L_{c_0} \neq \emptyset$. Since the output of the gate is 1, there must be an entry of the gate such that: either it is the ith entry of the circuit for some $1 \leq i \leq n$, and $w_i = 1$ (in which case, the two vertices $x_i, x'_i \in Lc_0$ are adjacent to both $a^3_{j_1}, b^3_{j_1}$ by construction); or it is the kth gate of the circuit for some $k < j_1$ and

this gate outputs 1. In the latter case, by minimality of j_1, $Y_k \cap L_{c_0} \neq \emptyset$. By construction, every vertex in Y_k is adjacent to $a_{j_1}^3, b_{j_1}^3$. As a result, $a_{j_1}^3, b_{j_1}^3$ have a neighbour in L_{c_0} in this subcase. The latter implies the existence of an edge with both endpoints in L_{c_0} since $N_{j_1} = \{a_{j_1}^3, b_{j_1}^3\}$. $\diamond$

3.3 Proof of Theorem 2

Proof of Theorem 2. Let $\langle \mathcal{C}, w \rangle$ be any instance of Monotone Circuit Value. Let $G = (V, E)$ be the graph obtained with our reduction from Section 3.1, which can be constructed in polylogarithmic-time with a polynomial number of processors. The graph G defines a coloring game. We fix any Nash equilibrium for this game, that exists by Theorem 1. By Claim 5, any two vertices in X have the same colour c_0. We will prove that there is at least one vertex in Y_m with colour c_0 if and only if the circuit $\mathcal{C}$ outputs 1 when it takes w as input. Since Monotone Circuit Value is PTIME-complete [9], the latter will prove that computing a Nash equilibrium for coloring games is PTIME-hard.

By Claim 6, we only need to prove that for every $1 \leq j \leq m$, there is at least one vertex in $Y_j \cup N_j$ with colour c_0. To prove it by contradiction, let j_0 be the smallest index j such that no vertex in $Y_j \cup N_j$ has colour c_0. By Claim 5, every vertex in $Y_{j_0} \cup N_{j_0}$ has the same colour as her private group. In particular, the three of $a_{j_0}^1, a_{j_0}^2, a_{j_0}^3$ receive payoff $2(n + j_0) - 3$. We will prove that one of these three agents could increase her payoff by choosing c_0 as her new colour, thus contradicting that we are in a Nash equilibrium. Indeed, by the minimality of j_0, it follows by Claim 4 that for any $1 \leq j \leq j_0 - 1$, there are exactly two vertices of $Y_j \cup N_j$ with colour c_0, while for every $j_0 \leq j \leq m$ there is no vertex in $Y_j \cup N_j$ with colour c_0. As a result, $|L_{c_0}| = 2(n + j_0) - 2$. In particular, any agent among $a_{j_0}^1, a_{j_0}^2, a_{j_0}^3$ could increase her payoff by choosing c_0 as her new colour — provided she is nonadjacent to every vertex in L_{c_0}. We will show it is the case for at least one of the three vertices, that will conclude the proof of the theorem. Assume w.l.o.g. that the j_0^{th} gate is an OR-gate (indeed, since by de Morgan's laws, the negation of an AND-gate can be transformed into an OR-gate and vice-versa, both cases are symmetrical). There are two cases.

Suppose that the output of the j_0^{th} gate is 1. In such case, there must be an entry of the gate such that: it is the ith entry of the circuit, for some $1 \leq i \leq n$, and $w_i = 1$; or it is the kth gate of the circuit for some $k < j_0$ and the output of that gate is 1. In the latter case, we have by Claim 6 that the two vertices of $Y_k \cup N_k$ with colour c_0 are in the set Y_k. Assume w.l.o.g. that the above-mentioned entry is the first entry of the gate. By construction, the two vertices $a_{j_0}^1, b_{j_0}^1$ are nonadjacent to every vertex in L_{c_0}. Else, the output of the j_0^{th} gate is 0. Therefore, for every entry of the gate: either it is the ith entry of the circuit, for some $1 \leq i \leq n$, and $w_i = 0$; or it is the kth gate of the circuit for some $k < j_0$ and the output of that gate is 0. In the latter case, we have by Claim 6 that the two vertices of $Y_k \cup N_k$ with colour c_0 are in the set N_k. By construction, the two vertices $a_{j_0}^3, b_{j_0}^3$ are nonadjacent to every vertex in L_{c_0}. In both cases, it contradicts that we are in a Nash equilibrium. $\square$

4 Conclusion and open perspectives

We suggest through this case study a more in-depth analysis of the complexity of computational mechanism designs. We would find it interesting to pursue similar investigations for other games. Experiments in the spirit of [11] could be helpful for our purposes. Further, we note that PRAM is seen by some as a too unrealistic model for parallel computation. Thus, one may argue that proving our reduction in this model casts a doubt on its reach. However, we can leverage on the stronger statement that MONOTONE CIRCUIT VALUE is *strictly* PTIME-hard [5]. It implies roughly that the sequential time and the parallel time to solve this problem cannot differ by more than a moderate polynomial-factor (unless the solving of *all* problems in PTIME can be sped up on a parallel machine by at least a polynomial-factor). Our reduction directly shows the same holds true for the problem of computing a Nash equilibrium in a given coloring game, that generalizes our hardness result to more recent parallel complexity classes (*e.g.*, [10]).

References

1. C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.
2. J. A. Bondy and U. S. R. Murty. *Graph theory*. Grad. Texts in Math., 2008.
3. N. Burani and W. S. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.
4. I. Chatzigiannakis, C. Koninis, P. N. Panagopoulou, and P. G. Spirakis. Distributed game-theoretic vertex coloring. In *OPODIS'10*, pages 103–118.
5. A. Condon. A theory of strict P-completeness. *Computational Complexity*, 4(3):220–241, 1994.
6. G. Ducoffe, D. Mazauric, and A. Chaintreau. The complexity of hedonic coalitions under bounded cooperation. Submitted.
7. J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *J. Comput. Syst. Sci.*, 63(1):21–41, 2001.
8. M. Gairing and R. Savani. Computing stable outcomes in hedonic games. In *SAGT'10*, pages 174–185.
9. R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to parallel computation: P-completeness theory*. Oxford University Press, 1995.
10. H. Karloff, S. Suri, and S. Vassilvitskii. A model of computation for MapReduce. In *SODA'10*, pages 938–948.
11. M. Kearns, S. Suri, and N. Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006.
12. J. Kleinberg and K. Ligett. Information-sharing in social networks. *Games and Economic Behavior*, 82:702–716, 2013.
13. J. R. Marden and A. Wierman. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.
14. P. N. Panagopoulou and P. G. Spirakis. A game theoretic approach for efficient graph coloring. In *ISAAC'08*, pages 183–195.
15. C. H. Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.
16. V. Vassilevska Williams. Fine-Grained Algorithms and Complexity (Invited Talk). In *STACS'16*.