Equilibria of Games in Networks for Local Tasks

Simon Collet[1]
CNRS and University Paris Diderot, France

Pierre Fraigniaud[2]
CNRS and University Paris Diderot, France

Paolo Penna
ETH Zurich, Switzerland

Abstract

Distributed tasks such as constructing a maximal independent set (MIS) in a network, or properly coloring the nodes or the edges of a network with reasonably few colors, are known to admit efficient distributed randomized algorithms. Those algorithms essentially proceed according to some simple generic rules, by letting each node choosing a temptative value at random, and checking whether this choice is consistent with the choices of the nodes in its vicinity. If this is the case, then the node outputs the chosen value, else it repeats the same process. Although such algorithms are, with high probability, running in a polylogarithmic number of rounds, they are not robust against actions performed by rational but selfish nodes. Indeed, such nodes may prefer specific individual outputs over others, e.g., because the formers suit better with some individual constraints. For instance, a node may prefer not being placed in a MIS as it is not willing to serve as a relay node. Similarly, a node may prefer not being assigned some radio frequencies (i.e., colors) as these frequencies would interfere with other devices running at that node. In this paper, we show that the probability distribution governing the choices of the output values in the generic algorithm can be tuned such that no nodes will rationally deviate from this distribution. More formally, and more generally, we prove that the large class of so-called LCL tasks, including MIS and coloring, admit simple "Luby's style" algorithms where the probability distribution governing the individual choices of the output values forms a Nash equilibrium. In fact, we establish the existence of a stronger form of equilibria, called symmetric trembling-hand perfect equilibria for those games.

2012 ACM Subject Classification Theory of computation → Distributed algorithms, Theory of computation → Algorithmic game theory, Theory of computation → Network games

Keywords and phrases Local distributed computing, Locally checkable labelings

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2018.6

1 Introduction

1.1 Motivation and Objective

In networks, independent sets and dominating sets can be used as backbones to collect, transfer, and broadcast information, and/or as cluster heads in clustering protocols (see, e.g., [19, 23]). Hence, a node belonging to some selected independent or dominating set may be subject to future costs in term of energy consumption, computational efforts, and bandwidth usage. As a consequence, rational selfish nodes might be tempted to deviate

[1] Funded by the European Research Council (ERC) under the H2020 research and innovation program (grant No 648032).
[2] Additional supports from the ANR project DESCARTES, and from the Inria project GANG.

from the instructions of an algorithm used to construct such sets, so that to avoid becoming member of the independent set, or dominating set, under construction. On the other hand, the absence of a backbone, or of cluster heads, may penalize the nodes. Hence every node is subject to a tension between (1) facilitating the *obtention* of a solution, and (2) avoiding certain *forms* of solutions.

A large class of randomized algorithms [5, 22] for constructing maximal independent sets (MIS) proceed in synchronous *rounds*, where a round allows every node to exchange information with its neighbors in the network, and to perform some individual computation. Roughly, at each round of these algorithms, every node i which has not yet decided applies to enter the MIS with a certain probability p_i. If a node applies to enter the MIS, and none of its neighbors simultaneously apply, then the former node enters the MIS, and, subsequently, all its neighbors renounce to enter the MIS. If two adjacent nodes simultaneously apply to enter the MIS, then there is a conflict, and both nodes remain undecided, and go to the next round. The round complexity of the algorithm heavily depends on the choice of the probability p_i that node i applies to enter the MIS, which may typically depend on the degree of node i, and may vary along with the execution of the algorithm, as node i accumulates more and more information about its neighborhood. Hence, a node i aiming at avoiding entering the MIS might be tempted to deviate from the algorithm by setting p_i small. However, if all nodes deviate in this way, then the algorithm may take a very long time before converging to a solution. The same holds whenever all nodes are aiming at entering the MIS.

Similar phenomenons may appear for other problems, like, e.g., coloring [7], that is, an abstraction of frequency assignment in radio networks. For solving this task, typical algorithms provides every node i with a probability distribution $\mathcal{D}_i$ over the colors, and node i chooses color c at random with probability $\mathcal{D}_i(c)$. If this color does not conflict with the chosen colors of its neighbors, then node i adopts this color, else it performs another random choice, and repeats until no conflicts with the neighbors occur. However, some frequencies might be preferred to others because, e.g., some frequencies might be conflicting with local devices present at the node. As a consequence, not all colors are equal for the nodes, and while each node is aiming at constructing a coloring quickly (in order to take benefit from the resulting radio network), it is also aiming at avoiding being assigned a color that it does not want. Therefore, in a random assignments of colors, every node might be tempted to give more weight to its preferred colors than to its non desired colors, and if all nodes deviate in this way, then the algorithm may take a long time before converging to a solution, if converging at all.

In fact, such phenomena as those listed above are susceptible to occur for many network problems, typically for solving so-called *locally checkable labeling* tasks [24], or LCL tasks for short.

1.1.1 Locally Checkable Labelings

LCL tasks [24] form a large class of classical problems, including maximal independent set, coloring, maximal matching, minimal dominating set, etc., studied for more than 20 years in the framework of distributed computing in networks. An LCL task is characterized by a finite set of *labels*, and a set of *good* labeled balls[3] of radius at most t, for some fixed $t \geq 0$.

[3] A *ball* of radius t is a graph with one identified node, called *center*, and with all the other nodes at distance at most t from the center. In a graph G, a ball of radius t centered at some node v is the subgraph induced by all nodes at distance at most t from v in G. A *labeled* ball is a ball whose every node is provided with a label (i.e., a bit-string).

For instance, in the MIS task, balls are of radius 1, a label is either $\bullet$ (interpreted as being member of the independent set), or $\circ$ (interpreted as not being member of the independent set), and a labeled ball is good if either (1) its center is labeled $\bullet$, and all its neighbors are labeled $\circ$, or (2) its center is labeled $\circ$, and at least one of its neighbors is labeled $\bullet$. Similarly, in k-coloring, the labels are in $\{1, \ldots, k\}$, balls are of radius 1, and a ball is good if the label of the center is different from the label of each of its neighbors.

Solving an LCL task consists in designing a distributed algorithm resulting in all nodes collectively assigning a label to each of them, such that all resulting balls are good.

In the following, we restrict ourselves to the large class of LCL tasks that are sequentially solvable by a greedy algorithm that (1) picks the nodes one by one in an arbitrary order, and (2) sets the label of each node when picked, after solely consulting the vicinity of the node. For instance, MIS is greedily constructible, as well as $(\Delta + 1)$-coloring in networks of maximum degree Δ. Instead, Δ-coloring is not greedily constructible, as witnessed by 2-coloring even cycles. We restrict ourselves to greedily constructible LCL tasks because non greedily constructible tasks are hard to handle in the distributed network computing setting. Indeed, for solving such tasks, far away nodes might be forced to coordinate, yielding poor locality, as witnessed by, again, 2-coloring even cycles (which cannot be solved in less than $\Omega(n)$ rounds [21]).

1.1.2 A Generic "Luby's Style" Randomized Algorithm for LCL Tasks

A generic randomized algorithm for LCL tasks, directly inspired from [5, 22], and therefore often referred to as "Luby's style" algorithm, performs as follows. Every node v aims at computing its label, *label*(v), for a given LCL task. The labels should be such that all resulting labeled balls are good with respect to the considered task. Node v starts with initial value *label*$(v) = \perp$. Let t be the radius of the task, i.e., the maximum radius of the labeled balls defining the tasks (both MIS and coloring have radius 1).

> **Distributed Construction Algorithm:** At each round, every node v which has not yet terminated observes the ball of radius t around it in the network (including its structure, and the already fixed nodes' labels). Then v chooses a random *temporary* label, *tmp-label*(v), compatible with the current fixed labels of the nodes in the observed ball. Next, v observes the ball of radius t around it in the network again, and recovers the temporary labels randomly chosen by the nodes in the ball. If *tmp-label*(v) is compatible with all fixed labels, and with all temporary labels in the ball of radius t centered at v (i.e., if the observed labeled ball in good w.r.t. the considered LCL task), then v sets *label*(v) as equal to *tmp-label*(v). The value of *label*(v) is then fixed, and it is not subject to any modification in the future. Otherwise, node v goes to the next round.

Note that assuming that the LCL task is greedily constructive prevents nodes from being blocked by nodes that terminated at previous rounds: every node has always at least one label at its disposal for building a good ball around it. (It can be easily checked that, for all greedily constructible LCL tasks, the generic distributed construction algorithm terminates, and outputs correctly, as long as every label has non zero probability to be chosen). Note that each phase of the generic algorithm takes only $2t$ rounds for LCL task of radius t, as every node performs two snapshots of the labels (fixed or temporary) in its t-ball.

The random choice of *tmp-label*(v) is governed by some probability distribution $\mathcal{D}$, which is actually characterizing the algorithm, and must be tuned according to the LCL task at hand. Importantly, $\mathcal{D}$ may depend on the round, and may also depend on the structure of

v's neighborhood as observed during the previous rounds. It is known that, for many tasks such as Maximal Independent Set (MIS), and $(\Delta + 1)$-coloring, there exist distributions $\mathcal{D}$ enabling the generic distributed construction algorithm to terminate in $O(\log n)$ rounds in n-node networks (see, e.g., [5, 7, 22]). In this paper, we consider the following issue:

What if selfish nodes are not playing according to the desired distribution $\mathcal{D}$?

To address this issue, we define *LCL games*.

1.1.3 LCL Games

To every LCL task can be associated a game, that we call *LCL game*, and that we define as follows. Let G be a connected simple graph. Every node v of G is a rational and selfish *player*, playing the game with the ultimate goal of maximizing its payoff while performing the generic distributed construction algorithm described in Section 1.1.2.

Strategy. A *strategy* for a node v is a probability distribution $\mathcal{D}$ over the labels compatible with the ball of radius t centered at v, which may depend on the history of v during the execution of the generic algorithm. For instance, in the MIS game, a strategy is a probability p to propose itself for entering the MIS. Similarly, in the $(\Delta + 1)$-coloring game, a strategy is a distribution of probabilities over the set of remaining colors compatible with the colors already assigned to the neighbors. (This set may even include a "fake" color 0 if nodes do not need to be systematically participating to a choice of color at every round [7]).

▶ Remark. The distribution $\mathcal{D}$ over the labels compatible with the ball of radius t centered at v is the unique item subject to non-orthodox behaviors. In particular, in LCL games, every node executes the prescribed algorithm, forwards messages correctly, and does not lie about its internal state, apart from what is concerning its private strategy for choosing its temporary label at random.

The *strategy* of a node at a round, i.e., the distribution $\mathcal{D}$ of probability over the labels, may depend on the history of that node at any point in time during the execution of the generic distributed construction algorithm. On the other hand, the individual strategies depend only on the knowledge accumulated by the nodes along with the execution of the algorithm. In fact, at the beginning of the algorithm, player v does not even know which node she will play in the network, and just knows that the network may belong to some given graph family (like, e.g., cycles, planar graphs, etc.).

Payoff. The *payoff* of the nodes is aiming at capturing the tension between the objective of every node to compute a global solution rapidly (as this global solution brings benefits to every node), versus avoiding certain forms of solutions (which may not be desirable from an individual perspective). We denote by $pref_v$ a preference function, which is an abstraction of how much node v will "suffer" in the future according to a computed solution. For instance, in the MIS game where nodes do not want to belong the constructed MIS, one could set

$$pref_v(I) = \begin{cases} 0 & \text{if } v \in I \\ 1 & \text{otherwise} \end{cases}$$

for every MIS I. More specifically, we define, for each node v,

$$pref_v : \{\text{good balls}\} \to [0, 1]$$

by associating to each good ball B centered at v the preference $pref_v(B)$ of v for that ball. The *payoff function* π_v of node v at the completion of the algorithm is decaying with the number k of rounds before the algorithm terminates at v. More precisely, we set

$$\pi_v = \delta^k \; pref_v(B_v)$$

where $0 < \delta < 1$ is a discount factor, B_v is the good ball centered at v as returned by the algorithm, and k is the number of rounds performed before all nodes in B_v fix their labels.

The choice of $\delta \in (0, 1)$ reflects the tradeoff between the quality of the solution from the nodes' perspective, and their desire to construct a global solution quickly. Note that k is at least the time it takes for v to fix its label, and at most the time it takes for the algorithm to terminate at all nodes. If the algorithm does not terminate around v, that is, if a label remains perpetually undecided in at least one node of B_v, then we set $\pi_v = 0$.

- The payoff of a node v will thus be large if all nodes in B_v decide quickly (i.e., k is small), and if the labels computed in B_v suits node v (i.e., $pref_v(B_v)$ is close to 1).
- Conversely, if v or another node in its ball B_v takes many rounds before deciding a label (i.e., k is large), or if node v is not satisfied by the computed solution in B_v (i.e., $pref_v(B_v)$ is close to 0), then the payoff of v will be small.

In particular, if the preference for every good ball is the same, then maximizing the payoff is equivalent to completing the task as quickly as possible. Instead, if the preference is very small for some balls, then nodes might be willing to slow down the completion of the task, with the objective of avoiding being the center of such a ball, in order to maximize their payoff. That is, such nodes may bias their distribution $\mathcal{D}$ towards preferred good balls, even if this is at the price of increasing the probability of conflicting with the choices of close nodes, resulting in more iterations before reaching convergence.

1.2 Our Results

We show that LCL games have *trembling-hand perfect equilibria*, that is, a stronger form of sequential equilibria due to Reinhard Selten [29], which are themselves a stronger form of Nash equilibria. Trembling-hand perfect equilibria include the possibility of off-the-equilibrium play, i.e., players may, with small probabilities, choose unintended strategies. In contrast, in Nash equilibria, players are assumed to play precisely as specified by the equilibrium. We show the following:

▶ **Theorem 1.** *For any greedily constructible locally checkable labeling, the LCL game associated to that labeling has a symmetric trembling-hand perfect equilibrium.*

Therefore, in particular, for many tasks occurring in the context of distributed network computing such as MIS, and $(\Delta + 1)$-coloring, there exist strategies played by the nodes of the network for solving these tasks such that no nodes have incentive to deviate from these strategies. Moreover, the related equilibria are strong forms of Nash equilibria which ensure that the players behave rationally even off the equilibrium path.

To establish Theorem 1, we first notice that LCL games belongs to the class of *extensive* games with *imperfect information*, because a node plays arbitrarily many times, and is not necessarily aware of the actions taken by far away nodes in the network. Also, LCL games belongs to the class of games with *infinite horizon* and *finite action set*: the horizon is infinite because neighboring nodes may perpetually prevent each other from terminating, and the action set is supposed to be finite (as long as the set of labels is finite). However, the classical game theoretical result [12] does not explicitly apply to LCL games. Indeed, first, in LCL games the actions of far-away nodes are not observable. Second, the imperfect information in [12] is solely related to the fact that players play simultaneously, while, again, in LCL games, imperfect information also refers to the fact that each node is not aware of the states of far away nodes in the network. It follows that the first step in our proof consists of revisiting the results in [12] for extending them, as specified in the following result.

▶ **Lemma 2.** *Every infinite, continuous, measurable, well-rounded, extensive game with perfect recall and finite action set has a trembling-hand perfect equilibrium. Moreover, if the game is, in addition, symmetric, then it has a symmetric trembling-hand perfect equilibrium.*

The hypotheses regarding the nature of the strategy, and the nature of the payoff function (continuity, measurability, etc.) are standard in the framework of extensive games. The notion of *well-rounded* game is new, and is used to capture the fact that the nodes play in synchronous rounds in LCL games. The fact that the equilibrium is symmetric is crucial as far as games in networks are concerned since, in LCL games, as in randomized distributed computing in general, the instructions given to all nodes are identical, and the behavior of the nodes only vary along with the execution of the algorithm when they progressively discover their environment. Extending the results in [12] is quite technical, but follows the standard methods for establishing such results in game theory. Therefore, we have chosen not to include the proof of Lemma 2 in this extended abstract.

The more interesting part of the proof, as far as local distributed computing in networks is concerned, is to show that LCL games satisfy all requirements stated in Lemma 2. This is the role of the following result:

▶ **Lemma 3.** *LCL games are symmetric, infinite, continuous, measurable, well-rounded, extensive games with perfect recall and finite action set.*

Lemmas 2 and 3 together prove Theorem 1. The rest of the paper is therefore focussing on formalizing LCL games, and on proving Lemma 3.

1.3 Related Work

Let us first position our result into the various settings of game theory. Indeed, games take various forms, and the types of equilibria that can be satisfied by these games vary according to their forms. Table 1 surveys the results regarding equilibria for various game settings, from the finite strategic games to the extensive games with imperfect information (we restrict our attention to games with a finite number of players). Recall that *trembling-hand perfect* equilibria [29] are refinements of *sequential* equilibria, which are themselves refinements of *subgame-perfect* equilibria, all of them being Nash equilibria. In Table 1, we distinguish strategic games (i.e., 1-step games like, e.g., prisoner's dilemma) from extensive games (i.e., game trees with payoffs, like, e.g., monetary policy in economy). For the latter class, we also distinguish games with perfect information (i.e., every player knows exactly what has taken place earlier in the game), from the games with imperfect information. We also distinguish finite games (i.e., games with a finite number of pure strategies, and finite number of repetitions) from games with infinite horizon (i.e., games which can be repeated infinitely often). The latter class of games is also split into games with finite numbers of actions, and games with infinite set of actions (like, e.g., when fixing the price of a product). In particular, Fudenberg and Levine [12] have proved that, under specific assumptions, every extensive game with imperfect information and finite action set has a sequential equilibrium (for behavior strategies). The specific class of games for which this result holds can be described as extensive games with observable actions, simultaneous moves, perfect recall, and finite action set, plus some continuity requirements. Although this class of games captures repeated games, and contains natural games in economy, LCL games are not explicitly included into this class. Indeed, as we already mentioned, the actions of far-away nodes are not observable

Table 1 A summary of results about the existence of equilibria.

	Strategic Games	**Extensive games with perfect information**	**Extensive games with imperfect information**
Finite games	[25] Nash equilibrium Mixed strategies	[28] Subgame-perfect equilibrium Pure strategies	[29] Trembling-hand perfect eq. Behavior strategies
Games with a finite action set		[12] Subgame-perfect equilibrium Pure strategies	[12] Sequential equilibrium Behavior strategies
Games with an infinite action set	[11] [14] Nash equilibrium Mixed strategies	[16] Subgame-perfect equilibrium Pure strategies	[10] Nash equilibrium Behavior strategies

in LCL games, and, in these latter games, imperfect information also refers to the fact that each node is not aware of the states of far away nodes in the network.

We now list some previous works related to games in networks (for network formation games, see, e.g., [6, 17]). Many games in networks have complete information, and, among games with incomplete information, a large part of the literature is dedicated to single-stage games where players are not initially aware of the network topology (see the survey [18]). Repeated games in networks have also been considered a lot in the literature (again, see [18]). These games differ from LCL games since, in repeated games, the utility of a player depends on each round, and it is computed pairwise with each neighbor, while, in LCL games, the utility is computed solely when the player terminates, and may depend on the whole neighborhood. Regarding games with incomplete information involving communications in networks, it is worth mentioning [2, 8, 9, 13, 15]. However, all these work mostly refer to games in which the players' actions consist in choosing which information to reveal, and to whom it should be revealed. Instead, in LCL games, players actions are always fully observable by their neighbors at distance $\leq t$, where t is the maximum radius of the good balls for the considered LCL task.

Probably the first contribution to distributed computing by rational agents is [1], which studies leader election in various networks, including complete networks and rings. Different forms of Nash equilibria are shown to exist, for both synchronous and asynchronous computing. The contribution in [3] extended and generalized the results in [1] by considering other problems (consensus, renaming, etc.), and by identifying different utility functions that encompass different preferences of players in a distributed system: communication preference, solution preference, and output preference. The paper [4] carried on this line of research, by enlarging the considered set of problems to coloring, partition, orientation, etc., and by addressing the question of how much global information agents should know a priori about the network in order for equilibria to exist. All these previous work differ from our approach in many ways. First, in [1, 3, 4], the agents strategies define the algorithm itself, including which messages to send, which information to reveal, etc. Instead, in this paper, the agents strategies solely consist in choosing a probability distribution on the possible outputs (at each round, depending on the history of the player). Second, the algorithms in the three aforementioned papers are "global" in the sense that they can take $\Omega(n)$ rounds in n-node networks. Instead, our (generic) algorithm is in essence "local", i.e., it is expected to converge in a polylogarithmic number of rounds, even in networks with large diameter. Last but not least, we consider a whole family of tasks at once (all "reasonable" LCL tasks) while the three aforementioned papers address each task separately, each one with its own algorithm.

2 The Extensive Games Related to LCL Games

In this section, we specify the type of games we are interested in, aiming at capturing the characteristics of LCL games. We focus on extensive games with imperfect information, and we include infinite horizon in the analysis of such games. We formally define all the concepts appearing in the statement of Lemma 2, and/or useful for formally defining LCL games, and proving Lemma 3. In particular, we define the novel notion of *well-rounded* games, which fits with distributed network computing in the LOCAL model [26].

2.1 Basic Definitions

Recall that an extensive game is a tuple $\Gamma = (N, A, X, P, U, p, \pi)$, where:

- $N = \{1, \ldots, n\}$ is the set representing the *players* of the game. An additional player, denoted by c, and called *chance*, is modeling all external random effects that might occur in the course of the game.
- A is the (finite) *action set*, i.e., a finite set representing the actions that can be made by each player when she has to play.
- X is the *game tree*, that is, a subset of $A^* \cup A^\omega$ where A^* (resp., A^ω) denotes the set of finite (resp., infinite) strings with elements in A, satisfying the following properties:
  - the empty sequence $\varnothing \in X$;
  - X is stable by prefix;
  - if $(a_i)_{i=1,\ldots,k} \in X$ for every $k \geq 1$, then $(a_i)_{i \geq 1} \in X$.

  The set X is partially ordered by the prefix relation, denoted by $\preceq$, where $x \preceq y$ means that x is a prefix of y, and $x \prec y$ means that x is a prefix of y distinct from y. The elements of X are called *histories*. A history x is *terminal* if it is a prefix of no other histories in X. In particular, every infinite history in X is terminal. The set of terminal histories is denoted by Z. If the longest history is finite then the game has *finite horizon*, otherwise it has *infinite horizon*. For every non-terminal history x, we denote by $A(x) = \{a \in A : (x, a) \in X\}$ the set of available actions after the history x.
- P is the *player partition*, i.e., a function $P : X \setminus Z \to N \cup \{c\}$ that assigns a player to each non-terminal history. $P(x)$ is the player who has to play after the history x. The sets $P_i = \{x \in X \setminus Z : P(x) = i\}$, for $i \in N \cup \{c\}$, called *player sets*, form a partition of $X \setminus Z$.
- U is the *information partition*, that is, a refinement of the player partition, whose elements are called *information sets*, such that for every $u \in U$, and for every two histories x, y in this *information set* u, we have $A(x) = A(y)$, i.e., the sets of available actions after x and after y are identical. We can therefore define $A(u)$ as the set of actions available after the information set u. Formally, $A(u) = \{a \in A : (x, a) \in X \text{ for every } x \in u\}$. For every history x, the information set containing x is denoted by $u(x)$. We also define $P(u)$ as the player who has to play after the information set u has been reached, and for every player i, the set $U_i = \{u \in U : P(u) = i\}$. The collection $\{U_i, i \in N \cup \{c\}\}$ forms a partition of U. Information sets regroup histories that are indistinguishable to players. Since the chance player c is not expected to behave rationally, we will simply put $U_c = \{\{x\}, x \in P_c\}$.
- p is a function that assigns to every history x in P_c (the player set of the chance c) a probability distribution over the set $A(x)$ of available actions after the history x. This *chance function* p is supposed to be common knowledge among the players.

- π is the *payoff function*, that is, $\pi : Z \to \mathbb{R}^n$ assigns the payoff (a real value) to every player in N for every terminal history of the game. We assume that every payoff is in $[-M, M]$ for some $M \geq 0$.

2.2 Well-Rounded Games

We introduce the concept of *rounds* in extensive games, and of *well-rounded* games.

▶ **Definition 4.** The *round function* r of an extensive game assigns a positive integer to every non terminal history x, defined by $r(x) = |Rec(x)|$ where

$$Rec(x) = \{x' \in X \mid x' \prec x \text{ and } P(x') = P(x)\}.$$

We call $r(x)$ the *round* of x. The round of a finite terminal history is the round of its predecessor, and the round of an infinite history z is $r(z) = \infty$. An extensive game Γ for which the round function is non decreasing with respect to the prefix relation, i.e.,

$$y \preceq x \Longrightarrow r(y) \leq r(x),$$

is said to be *well-rounded.*

Note that not every game is well-rounded, because two histories x and y such that $x \preceq y$ do not necessarily satisfy $P(x) = P(y)$. In a well-rounded game, since r is non decreasing, we have that, for any non terminal history x, every player has played at most $r(x) + 1$ times before x. Moreover, every player which has played less than $r(x)$ times before x will never play again after x.

Let $u \in U_i$ and $u' \in U_i$ be two (non necessarily distinct) information sets of the same player i, for which there exist $x \in u$, $x' \in u'$, and $a \in A(u')$, such that $(x', a) \preceq x$. Recall that an extensive game is said to have *perfect recall* if, for every such i, u, u' and a, we have:

$$\forall y \in u, \exists y' \in u' \mid (y', a) \preceq y.$$

The following lemma will allow us to safely talk about the round of an information set.

▶ **Lemma 5.** *Let Γ be an extensive game with perfect recall, and let $x \in X$ and $x' \in X$ be two non terminal histories in the same information set $u \in U$. Then x and x' have the same round.*

Proof. We first observe the following. Let Γ be an extensive game with perfect recall, and let $y \in X$ be a finite history. Let $y' \in X$ and $y'' \in X$ for which there exists $u \in U$ such that

$$y' \in Rec(y) \cap u \text{ and } y'' \in Rec(y) \cap u.$$

Then $y' = y''$. Indeed, since both y' and y'' are in $Rec(y)$, we have that both are prefixes of y, and thus one of the two is a prefix of the other. Assume, w.l.o.g., that $y'' \prec y' \prec y$ (as, if $y' = y''$ then we are done). Let a be the action such that $(y'', a) \preceq y'$. Since the game has perfect recall, there must exist a history $y''' \in u$ such that $(y''', a) \preceq y''$. Thus $y''' \prec y'' \prec y' \prec y$.We can repeat the same reasoning for y''' and y'' as we did for y'' and y'. In this way, we construct an infinite strictly decreasing sequence of histories, which contradicts the fact that y is finite.

If both $Rec(x)$ and $Rec(x')$ are empty, then x and x' have the same round. Assume, w.l.o.g., that $Rec(x) \neq \varnothing$, and let $y \in Rec(x)$. Let a be the action such that $(y, a) \preceq x$. Since the game has perfect recall, there exists $y' \in u(x')$ such that $(y', a) \preceq x'$. Therefore

$y' \prec x'$ and $P(y') = P(y) = P(x) = P(x')$. It follows that $y' \in Rec(x')$. Thus, for any $y \in Rec(x)$, we have identified a corresponding $y' \in Rec(x')$. This mapping from $Rec(x)$ to $Rec(x')$ is one-to-one. Indeed, let y_1 and y_2 in $Rec(x)$, and let y'_1 and y'_2 in $Rec(x')$ be the corresponding histories. If $y'_1 = y'_2$, then, since $u(y_1) = u(y'_1)$ and $u(y_2) = u(y'_2) = u(y'_1)$, we get that

$$y_1 \in Rec(x) \cap u(y'_1) \text{ and } y_2 \in Rec(x) \cap u(y'_1).$$

It follows from the above observation that $y_1 = y_2$. Thus the mapping is one-to-one, and hence $r(x) \leq r(x')$. It follows that we also have $Rec(x') \neq \varnothing$. Therefore, we can apply the same reasoning by switching the roles of x and x', which yields $r(x') \leq r(x)$. Thus $r(x) = r(x')$. ◀

2.3 Strategies, Outcomes, and Expected Payoff

In this section, we first recall several basic concepts about extensive games with perfect recall. Without loss of generality, we restrict our attention to *behavioral strategies* since such strategies are outcome-equivalent to mixed strategies thanks to [20]. The main objective of this section is to define the *expected payoff function*, which is novel as it is adapted to infinite games.

Recall that, for an information set u, the *local strategy* $b_{i,u}$ of a player i is a probability distribution over the set $A(u)$ of actions available given u. The set of local strategies of player i for u is denoted by $B_{i,u}$. The *behavioral strategy* b_i of a player i is a function which assigns a local strategy $b_{i,u}$ to every information set u of this player. The set of all behavioral strategies of player i is denoted by B_i. A *strategy profile* is a n-tuple of behavioral strategies, one for each player. The set of all strategy profiles is $B = \times_{i \in N} B_i$. For each player i, we denote by B_{-i} the set $\times_{j \neq i} B_j$. Since

$$B = B_i \times B_{-i} = \times_{i \in N} B_i,$$

a strategy profile b can be identified different ways, as $b = (b_i, b_{-i}) = (b_1, b_2, \ldots, b_n)$. If every player plays according to a strategy profile b, then the outcome of the game in entirely determined, in the sense that every history x has a probability $\rho_b(x)$ of being reached. For every strategy profile b, and every history $x = (a_i)_{i=1,\ldots,k}$ where $k \in \mathbb{N} \cup \infty$, the *realization probability* of x is defined by $\rho_b(\varnothing) = 1$, and

$$\rho_b(x) = \prod_{i=0}^{k-1} b_{P(x_i),u(x_i)}(a_{i+1})$$

where $x_0 = \varnothing$, and, for every positive $i \leq k$, $x_i = (a_1, a_2, \ldots, a_i)$. For the chance player c, we simply identify its strategy with the chance function p:

$$P(x_i) = c \Rightarrow b_{P(x_i),u(x_i)}(a_{i+1}) = p(x_i, a_{i+1}).$$

The function $\rho_b : X \to [0, 1]$ is called the *outcome* of the game under the strategy profile b. An outcome $\rho : X \to [0, 1]$ is *feasible* if and only if there exists a strategy profile b such that $\rho = \rho_b$. The set of feasible outcomes of Γ is denoted by O.

We are now ready to define the *expected payoff*. Note that, in a game with infinite horizon, there can be uncountably many terminal histories. Therefore the definition of the probability measure on Z requires some care. For any finite history x, let

$$Z_x = \{z \in Z \mid x \preceq z\}.$$

Note that Z_x might be uncountable. Let Σ be the σ-algebra on Z generated by all sets of the form Z_x for some finite history x. For each strategy profile b, the measure μ_b on Σ is defined by: for every set Z_x, $\mu_b(Z_x) = \rho_b(x)$. This definition ensures that μ_b is a probability measure because $\mu_b(Z) = \mu_b(Z_\varnothing) = \rho_b(\varnothing) = 1$.

▶ **Definition 6.** Let π be a payoff function that is measurable on Σ. The *expected payoff function* Π assigns a real value $\Pi(b)$ to every strategy profile $b \in B$, defined by

$$\Pi(b) = \int_\Sigma \pi \; d\mu_b \; .$$

Note that each component of the expected payoff function is bounded by M, where M is the upper bound on every payoff. A game Γ whose payoff function π is measurable on Σ is said to be a *measurable game*. In the following, we always assume that the considered games are measurable.

2.4 Equilibria, and Subgame Perfection

We now show how to adapt the standard notion of ϵ-equilibria (cf., e.g., [27]) to infinite games (Nash equilibria are special cases of ϵ-equilibria, with $\epsilon = 0$). Recall that a strategy profile b is a ϵ-equilibrium if and only if, for every player i, and every behavior strategy $b'_i \in B_i$ of this player, we have $\Pi_i(b'_i, b_{-i}) - \Pi_i(b) \le \epsilon$. Similarly, we recall the notions of *subgames* and *subgame perfect equilibria* (see, e.g., [28]). A subtree X' of X is said to be *regular* if no information sets contain both a history in X' and a history not in X'. To each regular subtree X' is associated a game $\Gamma' = (N, A, X', P', U', p', \pi')$, where P', U', p' and π' are the restrictions of P, U, p and π to X', called a *subgame*. The notions of outcomes and expected payoff functions for subgames follow naturally.

▶ **Definition 7.** A strategy profile b is a *subgame perfect ϵ-equilibrium* of an infinite game Γ if and only if, for every subgame Γ', the restriction of b to Γ' is an ϵ-equilibrium.

Note that a subgame perfect ϵ-equilibrium of Γ is an ϵ-equilibrium.

2.5 Metrics

In this section, we now define specific metrics on the set O of feasible outcomes, and on the set of behavior strategy profiles. These definitions are inspired from [12], with adaptations to fit our infinite setting.

▶ **Definition 8.** Let $\rho^1, \rho^2 \in O$ be two feasible outcomes of the same extensive game Γ. We define the following metric d on O: $d(\rho^1, \rho^2) = \sup_{x \in X, x \text{ finite}} 2^{-r(x)} \cdot \left|\rho^1(x) - \rho^2(x)\right|$ where $r(x)$ is the round of the finite history x.

▶ **Lemma 9.** *The function $d : O \times O \to \mathbb{R}$ specified in Definition 8 is a metric.*

Proof. We first show that d satisfies the triangle inequality. Let ρ^1, ρ^2 and ρ^3 be three feasible outcomes of Γ. For any finite history x we have the following:

$$\begin{array}{rll}
 & |\rho^1(x) - \rho^3(x)| & \le |\rho^1(x) - \rho^2(x)| + |\rho^2(x) - \rho^3(x)| \\
\Rightarrow & 2^{-r(x)}|\rho^1(x) - \rho^3(x)| & \le 2^{-r(x)}|\rho^1(x) - \rho^2(x)| + 2^{-r(x)}|\rho^2(x) - \rho^3(x)| \\
\Rightarrow & \sup\limits_{\substack{x \in X \\ x \text{ finite}}} 2^{-r(x)}|\rho^1(x) - \rho^3(x)| & \le \sup\limits_{\substack{x \in X \\ x \text{ finite}}} \left(2^{-r(x)}|\rho^1(x) - \rho^2(x)| + 2^{-r(x)}|\rho^2(x) - \rho^3(x)|\right) \\
\Rightarrow & \sup\limits_{\substack{x \in X \\ x \text{ finite}}} 2^{-r(x)}|\rho^1(x) - \rho^3(x)| & \le \sup\limits_{\substack{x \in X \\ x \text{ finite}}} 2^{-r(x)}|\rho^1(x) - \rho^2(x)| + \sup\limits_{\substack{x \in X \\ x \text{ finite}}} 2^{-r(x)}|\rho^2(x) - \rho^3(x)| \\
\Rightarrow & d(\rho^1, \rho^3) & \le d(\rho^1, \rho^2) + d(\rho^2, \rho^3).
\end{array}$$

Next, we prove that d separates different outcomes. Let ρ^1 and ρ^2 be two feasible outcomes such that $d(\rho^1, \rho^2) = 0$. By definition, this implies that, for every finite history x, we have $\rho^1(x) = \rho^2(x)$. Let b^1 and b^2 be two strategy profiles such that $\rho^1 = \rho_{b^1}$ and $\rho^2 = \rho_{b^2}$. Let $x = (a_k)_{k\geq 1}$ be an infinite history, and, for $k \geq 1$, let $x_k = (a_1, a_2, \ldots, a_k)$ be the corresponding increasing sequence of its prefixes. By definition, $\rho_{b^1}(x)$ is the limit, for $k \to \infty$, of the sequence $\rho_{b^1}(x_k)$, and $\rho_{b^2}(x_k) \to \rho_{b^2}(x)$ when $k \to \infty$ as well. Since the two sequences are equal, they have the same limit, and therefore $\rho^1(x) = \rho^2(x)$. Since this equality holds for every infinite history x, it follows that $\rho^1 = \rho^2$. ◀

We can use d to define a metric on behavioral strategy profiles as follows. Let $b^1, b^2 \in B$ be two behavioral strategy profiles of the same game Γ. We define the metric d on B by:

$$d(b^1, b^2) = \max\left\{ d\left(\rho_{b^1}, \rho_{b^2}\right),\ \sup_{\substack{i\in N\\ b_i\in B_i}} d\left(\rho_{(b_i, b^1_{-i})}, \rho_{(b_i, b^2_{-i})}\right) \right\}.$$

Finally, we define the continuity of the expected payoff function using the sup norm over $\mathbb{R}^n$. Specifically, the expected payoff function Π is *continuous* if, for every sequence of strategy profiles $(b^k)_{k\geq 1}$, and every strategy profile b, we have:

$$d(b^k, b) \underset{k\to\infty}{\to} 0 \implies \sup_{i\in N} \left|\Pi_i(b^k) - \Pi_i(b)\right| \underset{k\to\infty}{\to} 0.$$

An extensive game Γ is continuous if its expected payoff function is continuous.

2.6 Equilibria of Extensive Games Related to LCL Games

As stated in Lemma 2, it can be proved that every (symmetric) infinite, continuous, measurable, well-rounded, extensive game with perfect recall and finite action set has a (symmetric) trembling-hand perfect equilibrium. (Due to lack of space, this proof is not included in this extended abstract).

3 Proof of Lemma 3

In this section, we show that LCL games satisfy all hypotheses of Lemma 2, from which we derive Theorem 1. We start by formally defining LCL games.

3.1 Formal Definition of LCL Games

Let A be a finite alphabet, $\mathcal{F}$ a family of graphs with at most n vertices, $\mathcal{D}$ a probability distribution over $\mathcal{F}$, and $\mathcal{L}$ a greedily constructible LCL task over graphs in $\mathcal{F}$. Let $good(\mathcal{L})$ be the set of good balls in $\mathcal{L}$, and let t be the radius of $\mathcal{L}$, that is, the largest radius of the good balls. Let $pref_i : good(\mathcal{L}) \mapsto [0, 1]$, be a preference function of player i over good balls, and let $\delta \in (0, 1)$, called *discounting factor*. We define the game

$$\Gamma(\mathcal{L}, \mathcal{D}, pref, \delta) = (N, A, X, P, U, p, \pi)$$

associated to the LCL task $\mathcal{L}$, the distribution $\mathcal{D}$, the preference function $pref = (pref_i)_{i\in N}$, and the discounting factor δ, as follows.

- The player set is $N = \{1, \ldots, n\}$.
- The action set is $A \cup \mathcal{F}$ where the actions in $\mathcal{F}$ are only used by the chance player c in the initial move, and the actions in A are used by the actual players in N.

- The first move of the game is made by the chance player, that is, $P(\varnothing) = c$. As a result, a graph $G \in \mathcal{F}$ is selected at random according to the probability distribution $\mathcal{D}$, and a one-to-one mapping of the players to the nodes of G is chosen uniformly at random. From now on, the players are identified with the vertices of the graph G, labeled from 1 to n. Note that $\mathcal{F}$ might be reduced to a single graph, e.g., $\mathcal{F} = \{C_n\}$, and the chance player just selects, for each vertex v, which player $i \in N$ is playing at v (in a one-to-one manner).
- The game is then divided into rounds (corresponding to the intuitive meaning in synchronous distributed algorithms). At each round, the *active* players play in increasing order, from 1 to n. At round 0 every player is active and plays, and every action in A is available.
- At the end of each round (i.e., after every active player has played the same number of times), some players might become *inactive*, depending on the actions chosen during the previous rounds. For every $i \in N$, let $s(i)$ denote the last action played by player i, which we call the *state* of i, and let $ball(i)$ denote the ball of radius t centered at node i. Every player i such that $ball(i) \in good(\mathcal{L})$ at the end of a round becomes *inactive*.
- In subsequent rounds, the set of available actions might be restricted. For every round $r > 0$, and for every active player i, an action $a \in A$ is available to player i if and only if there exists a ball $b \in good(\mathcal{L})$ compatible with the states of inactive players in which $s(i) = a$.
- A history is terminal if and only if either it is infinite, or it comes after the end of a round with every player being inactive after that round.
- Let x be a history. We denote by $actions_i(x)$ the sequence of actions extracted from x by selecting all actions taken by player i during rounds before $r(x)$. (The action possibly made by player i at round $r(x)$, and actions made by a player $j \neq i$ are not included in $actions_i(x)$).
- Let x and y be two non terminal histories such that $P(x) = P(y) = i$. Then x and y are in the same information set if and only if, for every $j \in ball(i)$, we have

  $$actions_j(x) = actions_j(y).$$

  This can be interpreted by the fact that a player i "knows" every action previously taken by any player at distance at most t from i in the graph.
- Let i be a player, and let z be a terminal history. We define the *terminating time* of player i in history z by $time_i(z) = \max\{|actions_j(z)|, j \in ball(i)\} - 1$. The payoff function π of the game is then defined as follows. For every player i, and every terminal history z, we have $\pi_i(z) = \delta^{time_i(z)} \cdot pref_i(ball(i))$. And $\pi_i(z) = 0$ if $time_i(z) = \infty$.

3.2 The proof of Lemma 3

We survey the properties of LCL games, with emphasis on those listed as pre-conditions in the statement of Lemma 2.

▶ **Lemma 10.** *LCL games are well-rounded.*

Proof. This follows directly from the fact that, in a LCL game, (1) every active player plays at every round until it becomes inactive, and (2) once inactive, a player cannot become active again. ◀

▶ **Lemma 11.** *LCL games are symmetric.*

Proof. This follows directly from the fact that, in a LCL game, the position of every player in the actual graph (which might be fixed, or chosen at random in some given family of graphs according to some given distribution) is chosen uniformly at random. ◀

▶ **Lemma 12.** *LCL games have perfect recall.*

Proof. Let $\Gamma = (\mathcal{L}, \mathcal{D}, \mathit{pref}, \delta) = (N, A, X, P, U, p, \pi)$ be an LCL game. Let u and u' be two information sets of the same player i, for which there exists $x \in u$, $x' \in u'$, and $a \in A(u')$ such that $(x', a) \preceq x$. Let y be a history in u. Since x and y are in the same information set u, it follows that, for every player $j \in \mathit{ball}(i)$, we have $\mathit{actions}_j(x) = \mathit{actions}_j(y)$. In particular, this implies that x and y are in the same round. Let y' be the unique history which is a prefix of y with $P(y') = i$, and with $r(y') = r(x')$. (Such a history exists because $r(x') < r(x)$, and $r(y) = r(x)$). Since the players play in the same order at every round, we get that, for every player $j \in \mathit{ball}(i)$, $\mathit{actions}_j(x') = \mathit{actions}_j(y')$. As a consequence, we have $y' \in u'$. Furthermore, since $\mathit{actions}_i(x) = \mathit{actions}_i(y)$, the action played by i after y' must be a, which implies $(y', a) \preceq y$, and concludes the proof. ◀

▶ **Lemma 13.** *The payoff function π of a LCL game is measurable on the σ-algebra Σ corresponding to the game.*

Proof. We prove that, for every player i, and for every $a \in \mathbb{R}$, $\pi_i^{-1}(]a, +\infty[) \in \Sigma$, which implies that π is measurable on Σ. In LCL game, we have $\pi_i : Z \mapsto [0, 1]$. For every $a < 0$, we have $\pi_i^{-1}(]a, +\infty[) = Z \in \Sigma$. Similarly, for every $a > 1$, we have $\pi_i^{-1}(]a, +\infty[) = \varnothing \in \Sigma$. So, let us assume that $a \in]0, 1]$, and let z be a terminal history such that $\pi_i(z) > a$. We have $\mathit{time}_i(z) < \ln a / \ln \delta$, i.e., every player in $\mathit{ball}(i)$ has played only a finite number of times in the history z. Let x be the longest history such that $x \preceq z$, and $r(x) = \mathit{time}_i(z)$. By this setting, the history x' that comes right after x in z is the shortest prefix of z satisfying that every player in $\mathit{ball}(i)$ is inactive. Let z' be a terminal history such that $x' \preceq z'$. Since every player $j \in \mathit{ball}(i)$ is inactive after x', it follows that the state of any such player in z' is the same as its state in z, and thus $\pi_i(z') = \pi_i(z)$. It follows from the above that, for any terminal history z such that $\pi_i(z) > a$, there exists a finite history x' in round $\mathit{time}_i(z) + 1$ such that $z \in Z_{x'} \subseteq \pi_i^{-1}(]a, +\infty[)$. Since there are finitely many histories in round $\mathit{time}_i(z)$, we get that $\pi_i^{-1}(]a, +\infty[)$ is the union of a finite number of sets of the form $Z_{x'}$. As a consequence, it is measurable in Σ. It remains to prove that $\pi_i^{-1}(]0, +\infty[) \in \Sigma$. This simply follows from the fact that

$$\pi_i^{-1}(]0, +\infty[) = \bigcup_{k \geq 1} \pi_i^{-1}(]\frac{1}{k}, +\infty[),$$

and from the fact that Σ is stable by countable unions. ◀

▶ **Lemma 14.** *LCL games are continuous.*

Proof. Let b be a strategy profile, and let $(b^k)_{k \geq 0}$ be a sequence of strategy profiles such that $d(b^k, b) \to 0$ when $k \to \infty$. By definition of the metric d on B (cf. subsection 2.5), we have that $d(\rho_{b^k}, \rho_b) \to 0$ when $k \to \infty$. By definition of the metric on O, we have that, for any finite history x, $|\rho_{b^k}(x) - \rho_b(x)| \underset{k \to \infty}{\longrightarrow} 0$. It follows that, for any set of the form Z_x as defined in subsection 2.3, $|\mu_{b^k}(Z_x) - \mu_b(Z_x)| \underset{k \to \infty}{\longrightarrow} 0$. In other words the sequence of measures μ_{b^k} *strongly converges* to μ_b. Since, for every player i, the function π_i is measurable and bounded, it follows that

$$\int_\Sigma \pi_i \; d\mu_{b^k} \underset{k \to \infty}{\longrightarrow} \int_\Sigma \pi_i \; d\mu_b.$$

Therefore, $\Pi_i(b^k) \underset{k\to\infty}{\longrightarrow} \Pi_i(b)$, and thus the expected payoff function Π is continuous. ◀

Lemmas 10-14 show that every LCL game satisfies the requirements of Lemma 2, that is, every LCL game satisfies Lemma 3. ◀

4 Conclusion and Further Work

In this paper, we have proved that natural games occurring in the framework of local distributed network computing have trembling-hand perfect equilibria, a strong form of Nash equilibria. Further study includes the analysis of the performances of the robust algorithms resulting from these equilibria. This study is challenging as determining the performances of iterative distributed construction algorithms such as the generic algorithm in Section 1.1.2 is non trivial, even if nodes are altruistic, and follow the prescribed actions imposed by the algorithm. On the other hand, this line of study is of the utmost importance as, in the framework of large scale distributed computing, it is unreasonable to assume that no nodes will be tempted to deviate from the prescribed actions, for optimizing its own benefit, at the expense of the performances of the algorithms, and of the quality of the solutions.

References

1. Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. Distributed Protocols for Leader Election: A Game-Theoretic Perspective. In *27th International Symposium on Distributed Computing (DISC)*, pages 61–75, 2013. `doi:10.1007/978-3-642-41527-2_5`.
2. Daron Acemoglu, Munther A Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian learning in social networks. *The Review of Economic Studies*, 78(4):1201–1236, 2011.
3. Yehuda Afek, Yehonatan Ginzberg, Shir Landau Feibish, and Moshe Sulamy. Distributed computing building blocks for rational agents. In *Symposium on Principles of Distributed Computing (PODC)*, pages 406–415. ACM, 2014.
4. Yehuda Afek, Shaked Rafaeli, and Moshe Sulamy. Cheating by Duplication: Equilibrium Requires Global Knowledge. Technical report, arXiv, 2017. `arXiv:1711.04728`.
5. Noga Alon, László Babai, and Alon Itai. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. *J. Algorithms*, 7(4):567–583, 1986. `doi:10.1016/0196-6774(86)90019-2`.
6. Chen Avin, Avi Cohen, Pierre Fraigniaud, Zvi Lotker, and David Peleg. Preferential Attachment as a Unique Equilibrium. In *The Web Conference (WWW)*, New-York, 2018. ACM.
7. Leonid Barenboim and Michael Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2013.
8. A. Calvó-Armengol and J. de Martí Beltran. Information gathering in organizations: Equilibrium, welfare and optimal network structure. *Journal of the European Economic Association*, 7:116–161, 2009.
9. A. Calvó-Armengol, J. de Martí Beltran, and Prat A. Communication and influence. *Theoretical Economics*, 10:649–690, 2015.
10. Subir K Chakrabarti. Equilibrium in Behavior Strategies in Infinite Extensive Form Games with Imperfect Information. *Economic Theory*, 2(4):481–494, 1992.
11. Ky Fan. Fixed-Point and Minimax Theorems in Locally Convex Topological Linear Spaces. *PNAS*, 38(2):121–126, 1952.
12. Drew Fudenberg and David Levine. Subgame-Perfect Equilibria of Finite- and Infinite-Horizon Games. *Journal of Economic Theory*, 31(2):251–268, 1983.

13 Andrea Galeotti, Christian Ghiglino, and Francesco Squintani. Strategic information transmission networks. *J. Economic Theory*, 148(5):1751–1769, 2013.

14 Irving L. Glicksberg. A Further Generalization of the Kakutani Fixed Point Theorem with Application to Nash Equilibrium Points. *Proceedings of the AMS*, 3(1):170–174, 1952.

15 J. Hagenbach and F. Koessler. Strategic communication networks. *Review of Economic Studies*, 77(3):1072–1099, 2011.

16 Christopher Harris. Existence and Characterization of Perfect Equilibrium in Games of Perfect Information. *Econometrica: Journal of the Econometric Society*, 53(3):613–628, 1985.

17 Matthew O. Jackson, Brian W. Rogers, and Yves Zenou. Networks: An Economic Perspective. Technical report, arXiv, 2016.

18 Matthew O Jackson and Yves Zenou. Games on networks. In *Handbook of game theory with economic applications*, volume 4, pages 95–163. Elsevier, 2015.

19 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Radio Network Clustering from Scratch. In *12th European Symposium on Algorithms (ESA)*, LNCS 3221, pages 460–471. Springer, 2004.

20 Harold W. Kuhn. Extensive Games and the Problem of Information. In *Contributions to the Theory of Games*, volume II (Annals of Mathematics Studies, 28), pages 193–216. Princeton University Press, 1953.

21 Nathan Linial. Locality in Distributed Graph Algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

22 Michael Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986. doi:10.1137/0215074.

23 Thomas Moscibroda and Roger Wattenhofer. Maximal independent sets in radio networks. In *24th Symposium on Principles of Distributed Computing (PODC)*, pages 148–157. ACM, 2005. doi:10.1145/1073814.1073842.

24 Moni Naor and Larry J. Stockmeyer. What Can be Computed Locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

25 John F. Nash. Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

26 David Peleg. *Distributed Computing: A Locality-Sensitive Approach.* Discrete Mathematics and Applications. SIAM, Philadelphia, 2000.

27 Roy Radner. Collusive Behavior in non-Cooperative epsilon-Equilibria of Oligopolies with Long but Finite Lives. *Journal of Economic Theory*, 22:136–154, 1980.

28 Reinhard Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragetraheit. *Z. Gesamte Staatwissenschaft*, 12:301–324, 1965.

29 Reinhard Selten. Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games. *International journal of game theory*, 4(1):25–55, 1975.