Problèmes vérifiables par agents mobiles

Evangelos Bampas and David Ilcinkas

CNRS & Univ. Bordeaux, LaBRI, UMR 5800, F-33400, Talence, France

Nous considérons les problèmes de décision qui sont résolus de manière répartie par des agents mobiles synchrones évoluant dans un réseau inconnu et anonyme. Chaque agent dispose d'un identifiant unique et d'une chaîne d'entrée, et ces agents doivent décider collectivement de la validité d'une propriété qui peut être basée sur les chaînes d'entrée, le graphe dans lequel les agents évoluent, et leurs positions de départ. Poursuivant un travail récent de Friagniaud et Pelc [LATIN 2012, LNCS 7256, p. 362–374], nous introduisons plusieurs nouvelles classes naturelles de calculabilité par agents mobiles, nous permettant d'obtenir une classification plus fine des problèmes inclus dans co-MAV ou MAV, cette dernière étant la classe des problèmes vérifiables lorsque les agents disposent d'un certificat approprié. Dans cet article, nous exhibons des résultats d'inclusion ou de séparation entre toutes ces classes. Nous déterminons également leurs propriétés de clôture vis-à-vis des opérations classiques de la théorie des ensembles. Notre principal outil technique, intéressant en soi, est un nouveau méta-protocole qui permet l'exécution essentiellement en parallèle d'un nombre potentiellement infini de protocoles d'agents mobiles, de façon similaire à la technique classique de déployeur universel (dovetailing) présente en calculabilité classique.

Keywords: calcul par agents mobiles, classes de calculabilité, propriétés de clôture, déployeur universel (dovetailing)

1 Introduction

Recent years have seen a surge of research interest in the direction of putting forth computability and complexity theories for various aspects of distributed computing. Significant examples of this trend include the investigation of unreliable failure detectors [2], as well as wait-free hierarchies [8]. A more recent line of work studies the impact of randomization, non-determinism, and identifiers in what concerns the computational capabilities of the $\mathcal{LOCAL}$ model [3, 4, 5]. A different approach considers the characterization of problems that can be solved under various notions of termination detection or various types of knowledge about the network in message-passing systems [1, 7]. Finally, a recent work focuses on the computational power of teams of mobile agents [6]. Our work lies in this latter direction.

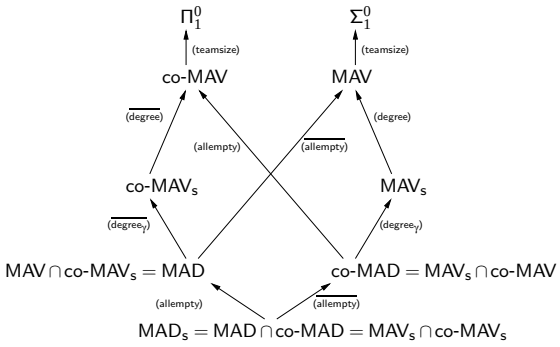
More specifically, we consider the setting of a distributed system in which the computation is performed by one or several deterministic mobile agents, operating in an unknown, anonymous, synchronous network. As usual in anonymous networks, we assume that on each node there is a local port labeling (from 1 up to the degree of the node) of its incident edges. Each agent has a unique identifier and is provided with an input string, and they have to collectively decide a property which may involve their input strings, the graph on which they are operating, and their particular starting positions. The execution proceeds in synchronous steps, in the beginning of each of which each agent becomes aware of the port number of the edge through which it entered the current node, the degree of the node, and the configurations of other agents on the same node. Then, the agent performs a local computation and decides whether to halt (by *accepting* or *rejecting*) or to move to a neighboring node (or stay put). We assume that all local computations take the same amount of time. The decision problems that the agents are required to solve are sets of triples of the form $(G,\vec{s},\vec{x})$, where G is a port labeled graph, $\vec{s}$ is a list of nodes representing the starting positions of the agents, and $\vec{x}$ is the list of input strings of the agents.

In [6], Fraigniaud and Pelc introduced two natural computability classes, MAD and MAV, as well as their counterparts co-MAD and co-MAV (the classes of all problems whose complements are in MAD and MAV, respectively). The class MAD, for "Mobile Agent Decidable", is the class of all mobile agent decision problems which can be decided by teams of mobile agents, in the sense that there exists a mobile agent protocol such that in "yes" instances all agents accept, while in "no" instances at least one agent rejects.

	Definition		*Closure Properties*		
	"yes" instances	"no" instances	Union	Intersection	Complement
$\mathsf{MAD_s}$	($\forall$ certificate :) yes	($\forall$ certificate :) no	✓	✓	✓
MAD	($\forall$ certificate :) yes	($\forall$ certificate :) $\widehat{\mathsf{no}}$	✗	✓	✗
co-MAD	($\forall$ certificate :) $\widehat{\mathsf{yes}}$	($\forall$ certificate :) no	✓	✗	✗
$\mathsf{MAV_s}$	$\exists$ certificate : yes	$\forall$ certificate : no	✓	✓	✗
$\mathsf{co\text{-}MAV_s}$	$\forall$ certificate : yes	$\exists$ certificate : no	✓	✓	✗
MAV	$\exists$ certificate : yes	$\forall$ certificate : $\widehat{\mathsf{no}}$	✗	✓	✗
co-MAV	$\forall$ certificate : $\widehat{\mathsf{yes}}$	$\exists$ certificate : no	✓	✗	✗
MAV′	$\exists$ certificate : $\widehat{\mathsf{yes}}$	$\forall$ certificate : no	✓	✓	✗
co-MAV′	$\forall$ certificate : yes	$\exists$ certificate : $\widehat{\mathsf{no}}$	✓	✓	✗

TABLE 1: Overview of mobile agent decidability and verifiability classes. The notation yes (resp. no) means that all agents accept (resp. reject). Similarly, $\widehat{\mathsf{yes}}$ (resp. $\widehat{\mathsf{no}}$) means that at least one agent accepts (resp. rejects).

FIGURE 1: Containments between classes below MAV and co-MAV. All inclusions are provably strict. Separating problems are displayed next to the arrows.



The class MAV, for "Mobile Agent Verifiable", is the class of all mobile agent decision problems which can be verified by teams of mobile agents. This means that, in a "yes" instance, there must exist a certificate for each agent such that if the agent receives it alongside its input string, then all agents must accept, whereas in a "no" instance, for every possible certificate, at least one agent must reject. Note that the certificates of the agents must be independent of the agent identifiers. Fraigniaud and Pelc proved in [6] that MAD is strictly included in MAV, and they exhibited a problem which is complete for MAV under an appropriate notion of oracle reduction.

The contributions of our paper can be summarized as follows :

— We introduce several new mobile agent computability classes which play a key role in our endeavor for a finer classification of problems below MAV and co-MAV. The classes $\mathsf{MAD_s}$ and $\mathsf{MAV_s}$ are strict versions of MAD and MAV, respectively, in which unanimity is required in both "yes" and "no" instances. Furthermore, we consider the class co-MAV′ of mobile agent decision problems that admit a certificate for "no" instances, while retaining the system-wide acceptance mechanism of MAV. Table 1 contains an overview of the classes we consider.

— We perform a thorough investigation of the relationships between the newly introduced and pre-existing classes. In particular, for every pair A, B of classes, we either prove that A is included in B, or we separate them by exhibiting a mobile agent decision problem that belongs to A but not to B. This leads to the class diagram of Figure 1.

— We complement our results with a complete study of the closure properties of these classes under the standard set-theoretic operations of union, intersection, and complement. These properties are summarized in Table 1.

— The main technical tool which we use to obtain these results is a new meta-protocol which is of independent interest, as it enables the execution of a possibly infinite number of mobile agent protocols essentially in parallel. This can be seen as a mobile agent computing analogue of the well-known dovetailing technique from classical recursion theory.

2 Mobile Agent Computability Classes

Decidability classes. By definition, $\mathsf{MAD_s}$ is a subset of both MAD and co-MAD and it is easy to check that $\mathsf{MAD_s} = \text{co-}\mathsf{MAD_s}$. Moreover, all of these classes are subsets of Δ_1^0 (the class of centrally decidable problems), since a centralized algorithm, provided with an encoding of the graph and the starting positions, inputs, and IDs of the agents, can simulate the corresponding mobile agent protocol and decide appropriately. As mentioned in [6], $\mathsf{path} = \{(G,\vec{s},\vec{x}) : G \text{ is a path}\}$ is an example of a decision problem which is in $\Delta_1^0 \setminus \mathsf{MAD}$, since, intuitively, an agent cannot distinguish a long path from a cycle. In fact, this observation yields $\mathsf{path} \in \Delta_1^0 \setminus (\mathsf{MAD} \cup \text{co-}\mathsf{MAD})$.

A nontrivial problem in $\mathsf{MAD_s}$ is $\mathsf{treesize} = \{(G,\vec{s},\vec{x}) : G \text{ is a tree of size } n \text{ and } \forall i\ x_i = n\}$. The problem was already shown to be in MAD in [6], and the protocol given there can be modified to show the stronger property that $\mathsf{treesize} \in \mathsf{MAD_s}$. However, it is not always possible to strengthen a MAD protocol to yield a $\mathsf{MAD_s}$ protocol. This is witnessed, for example, by the problem $\mathsf{allempty} = \{(G,\vec{s},\vec{x}) : \forall i\ x_i = \varepsilon\}$, which separates MAD from $\mathsf{MAD_s}$: $\mathsf{allempty} \in \mathsf{MAD} \setminus \mathsf{MAD_s}$. As a corollary, $\overline{\mathsf{allempty}} \in \text{co-}\mathsf{MAD} \setminus \mathsf{MAD_s}$ and thus we obtain a separation between co-MAD and $\mathsf{MAD_s}$. As we mentioned, $\mathsf{MAD_s}$ is included in both MAD and co-MAD. In fact, $\mathsf{MAD_s} = \mathsf{MAD} \cap \text{co-}\mathsf{MAD}$. This can be obtained as a corollary of Theorems 2 and 3.

Theorem 1 $\mathsf{MAD_s} = \mathsf{MAD} \cap \text{co-}\mathsf{MAD}$.

By Theorem 1, if allempty was included in co-MAD, we would obtain $\mathsf{allempty} \in \mathsf{MAD_s}$, which we know to be false. Thus, $\mathsf{allempty} \notin \text{co-}\mathsf{MAD}$ and we obtain a separation between MAD and co-MAD. Symmetrically, $\overline{\mathsf{allempty}} \in \text{co-}\mathsf{MAD} \setminus \mathsf{MAD}$.

Verifiability classes. By definition, $\mathsf{MAV_s} \subseteq \mathsf{MAV}$. Moreover, $\mathsf{MAV} \subseteq \Sigma_1^0$ (Σ_1^0 is the class of all semi-decidable problems, i.e., in a "no" instance the algorithm may not terminate), since a centralized algorithm can simulate the MAV protocol for all possible certificate vectors (by classical dovetailing) and accept if it finds a certificate for which all agents accept. This yields immediately that $\text{co-}\mathsf{MAV_s} \subseteq \text{co-}\mathsf{MAV} \subseteq \Pi_1^0 = \text{co-}\Sigma_1^0$.

The class MAV is, in fact, a strict superset of $\mathsf{MAV_s}$, since we show that, for the problem $\mathsf{degree} = \{(G,\vec{s},\vec{x}) : \forall i\ \exists v\ d_v = x_i\}$, we have $\mathsf{degree} \in \mathsf{MAV} \setminus \mathsf{MAV_s}$. As a corollary, $\overline{\mathsf{degree}} \in \text{co-}\mathsf{MAV} \setminus \text{co-}\mathsf{MAV_s}$. In order to separate Σ_1^0 from MAV and Π_1^0 from co-MAV, we observe that $\mathsf{teamsize} = \{(G,\vec{s},\vec{x}) : \forall i\ x_i = |\vec{s}|\}$, which is clearly in $\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$, is neither in MAV nor in co-MAV.

Decision problems with "no" certificates. In classical computability, the class $\Pi_1^0 = \text{co-}\Sigma_1^0$ can be seen as the class of problems that admit a "no" certificate, i.e. : for "no" instances, there exists a certificate that leads to rejection, whereas for "yes" instances, no certificate can lead to rejection. In this respect, while MAV can certainly be considered as the mobile agent analogue of Σ_1^0, co-MAV is not quite the analogue of Π_1^0. Problems in co-MAV indeed admit a "no" certificate, but the acceptance mechanism is reversed : for "no" instances, there exists a certificate that leads all agents to reject. This motivates us to define and study co-MAV′, the class of mobile agent problems that admit a "no" certificate while retaining the MAV acceptance mechanism, as well as its complement MAV′. By definition, we have that $\mathsf{MAV_s} \subseteq \mathsf{MAV}'$ and $\text{co-}\mathsf{MAV_s} \subseteq \text{co-}\mathsf{MAV}'$. In fact, we show that these hold as equalities :

Theorem 2 $\mathsf{MAV}' = \mathsf{MAV_s}$ *and* $\text{co-}\mathsf{MAV}' = \text{co-}\mathsf{MAV_s}$.

Connections with the decidability classes. From the definitions, we deduce immediately that $\mathsf{MAD} \subseteq \text{co-}\mathsf{MAV}'$, therefore, by Theorem 2, $\mathsf{MAD} \subseteq \text{co-}\mathsf{MAV_s}$. Similarly, $\text{co-}\mathsf{MAD} \subseteq \mathsf{MAV_s}$. Therefore, since $\mathsf{MAD_s} \subseteq \mathsf{MAD} \cap \text{co-}\mathsf{MAD}$, we also have that $\mathsf{MAD_s} \subseteq \mathsf{MAV_s} \cap \text{co-}\mathsf{MAV_s}$. In fact, we obtain the following result :

Theorem 3 $\mathsf{MAD_s} = \mathsf{MAV_s} \cap \text{co-}\mathsf{MAV_s}$.

It turns out that we can characterize in a similar way the classes MAD and co-MAD :

Theorem 4 $\mathsf{MAD} = \mathsf{MAV} \cap \text{co-}\mathsf{MAV_s}$ *and* $\text{co-}\mathsf{MAD} = \mathsf{MAV_s} \cap \text{co-}\mathsf{MAV}$.

It was shown in [6] that, for the restrictions $\mathsf{MAD_1}$ (resp. $\mathsf{MAV_1}$) of MAD (resp. MAV) to problems decidable (resp. verifiable) by a single agent, it holds that $\mathsf{MAD_1} = \mathsf{MAV_1} \cap \text{co-}\mathsf{MAV_1}$. Theorems 3 and 4 can be seen as generalizations of that result to multiagent classes.

For any fixed $\gamma \geq 1$, let $\mathsf{degree}_\gamma = \{(G,\vec{s},\vec{x}) : \exists v\ d_v = \gamma\}$. It holds that $\mathsf{degree}_\gamma \in \mathsf{MAV_s} \setminus \mathsf{co\text{-}MAD}$ and $\overline{\mathsf{degree}_\gamma} \in \mathsf{co\text{-}MAV_s} \setminus \mathsf{MAD}$. In view of Theorem 4, this yields a separation between $\mathsf{MAV_s}$ and co-MAV, as $\mathsf{degree}_\gamma \in \mathsf{MAV_s} \setminus \mathsf{co\text{-}MAV}$, and a separation between $\mathsf{co\text{-}MAV_s}$ and MAV, as $\overline{\mathsf{degree}_\gamma} \in \mathsf{co\text{-}MAV_s} \setminus \mathsf{MAV}$.

3 Simulated Parallel Execution of Mobile Agent Protocols

The key technical tool for the proofs of Theorems 1–4 is a new method for executing a (possibly infinite) number of mobile agent protocols essentially in parallel. This is similar in spirit to dovetailing in classical computing, except that the situation is considerably more complicated when mobile agents are concerned, due to the fact that there are several agents computing in an unknown graph, each knowing only part of the input. The meta-protocol that enables this parallel execution of protocols is parameterized by $\mathcal{N}, f, g$. The set $\mathcal{N}$ is a possibly infinite, recursively enumerable set of mobile agent protocols. The functions f and g are computable functions which represent local computations. The function f (global decider) receives as input the full instance including the agent identifiers and decides irrevocably (the agent either accepts or rejects). The function g (local decider) receives as input an arbitrary-length list of histories of executions of some of the protocols in $\mathcal{N}$ for a certain number of steps, as experienced by a particular agent, and it may accept, reject, or refuse to decide. The local decider must satisfy the property that, for sufficiently large number of steps and for sufficiently large number of histories, there will be a decisive output from g.

The meta-protocol works in iterated phases, which correspond to increasing values of a variable T that represents an assumed upper bound on the total execution time, the length of agent identifiers, and the number of nodes in the graph. In each phase, each agent first attempts to explore the $2T$-ball around its starting node. If it meets an agent with a lexicographically larger identifier, it continues to execute the current phase but enters a special *neutralized* state, meaning that at the end of the current phase it will become idle and wait for some other agent to provide it with the full knowledge of the current instance so as to decide via the global decider. On the other hand, if it meets a halted or idle agent, it becomes a *mapseeker*. Mapseekers attempt subsequently to construct a full map of the network, treating the halted or idle agent they located as a fixed mark. If they succeed, they then inform all agents about the full instance (including agent identifiers) and they all decide via the global decider. If not, then each agent executes T steps of the first T protocols in $\mathcal{N}$, returning each time back to its starting node. If, during the $2T$-ball exploration at the beginning of the phase, the agent did not meet any other agent and if it did not find any node of degree greater than T and, finally, if its own identifier is not longer than T bits, then the agent feeds the histories of these T-step executions into the local decider and either halts or continues to the next phase.

Références

[1] Boldi, P., Vigna, S. : An effective characterization of computability in anonymous networks. In : DISC 2001. LNCS, vol. 2180, pp. 33–47. Springer (2001)

[2] Chandra, T.D., Toueg, S. : Unreliable failure detectors for reliable distributed systems. J. ACM 43(2), 225–267 (1996)

[3] Fraigniaud, P., Göös, M., Korman, A., Parter, M., Peleg, D. : Randomized distributed decision. Distributed Computing 27(6), 419–434 (2014)

[4] Fraigniaud, P., Göös, M., Korman, A., Suomela, J. : What can be decided locally without identifiers ? In : PODC 2013. pp. 157–165. ACM (2013)

[5] Fraigniaud, P., Korman, A., Peleg, D. : Towards a complexity theory for local distributed computing. J. ACM 60(5), 35 (2013)

[6] Fraigniaud, P., Pelc, A. : Decidability classes for mobile agents computing. In : LATIN 2012. LNCS, vol. 7256, pp. 362–374. Springer (2012)

[7] Godard, E., Métivier, Y., Tel, G. : Termination Detection of Local Computations. arXiv :1001.2785 [cs.DC] (2010)

[8] Herlihy, M. : Wait-free synchronization. ACM Trans. Program. Lang. Syst. 13(1), 124–149 (1991)