On the Deterministic Tracking of Moving Objects with a Binary Sensor Network

Yann Busnel, Leonardo Querzoni, Roberto Baldoni, Marin Bertier,
Anne-Marie Kermarrec

On the Deterministic Tracking of Moving Objects with a Binary Sensor Network

Yann Busnel[1], Leonardo Querzoni[2], Roberto Baldoni[2], Marin Bertier[3], and Anne-Marie Kermarrec[4]

[1] IRISA / University of Rennes 1 – France
[2] MIDLAB / University of Rome – "La Sapienza" – Italy
[3] IRISA / INSA Rennes – France
[4] INRIA Rennes – Bretagne Atlantique – France

Abstract. This paper studies the problem of associating deterministically a track revealed by a binary sensor network with the trajectory of a unique moving anonymous object, namely the *Multiple Object Tracking and Identification* (MOTI) problem. In our model, the network is represented by a sparse connected graph where each vertex represents a binary sensor and there is an edge between two sensors if an object can pass from a sensed region to another without activating any other remaining sensor. The difficulty of MOTI lies in the fact that trajectories of two or more objects can be so close (track merging) that the corresponding tracks on the sensor network can no longer be distinguished, thus confusing the deterministic association between an object trajectory and a track.
The paper presents several results. We first show that MOTI cannot be solved on a general graph of ideal binary sensors even by an omniscient external observer if all the objects can freely move on the graph. Then, we describe some restrictions that can be imposed *a priori* either on the graph, on the object movements or both, to make MOTI problem always solvable. We also discuss the consequences of our results and present some related open problems.

1 Introduction

Context "*Tracking the movements of anonymous objects and associating each object trajectory with an unique identifier*" is a basic problem in many applicative contexts such as surveillance [8], rescue, traffic monitoring [3], pursuit evasion games, *etc.* Tracking objects with a sensor network is a challenging task. Potential inaccuracy of sensors (*e.g.* [1]) and the complexity of the localization subsystem computability (*e.g.* [2]) significantly complicate tracking initiation, maintenance and termination of object trajectories leading to false detection and missing observations.

Even without considering false detections and missing observations, the problem of associating an unique identifier to a track corresponding to the trajectory of one object is difficult due to the potential merging of tracks. This may confuse this association if sensors do not have adequate capabilities [7]. In other words, two tracks may be so close to each other at a certain time that they become indistinguishable, and then impossible to identify after splitting. Deciding, after a track merging, what is a relevant association among multiple hypothesis is usually achieved by looking at the behaviors of tracks

before the merging happened [7]. In the following, we refer to this one-to-one mapping between tracks and object trajectories as the *Multiple Object Tracking and Identification* (MOTI) problem.

We investigate MOTI by using a binary sensor network, which means that each sensor reports only a binary value indicating if an object is in its sensing region or not. The interest in focusing on such minimalist and simple devices is motivated by the fact that we want to study the essence of MOTI's solvability without looking at the powerfulness of sensors capabilities. As in [8], in our model, the sensor network is represented by a sparse connected graph, namely the *passage connectivity graph* (PCG), where each vertex represents a binary sensor and there is an edge between two sensors u and v only if an object can pass from the sensing region of u to the one of v without activating any other remaining sensor.

Contribution In this paper, we show that it is impossible to solve the MOTI problem in a generic graph. To make the impossibility as strong as possible, this last is proved considering an omniscient observer that has complete knowledge of the state of the graph; moreover, we assume ideal conditions for object tracking such as perfect binary sensors (*i.e.* no false detections or missing observations), ideal coverage of the sensing areas (*i.e.* disjoint sensing regions), so that at each instant of time one object activates only one sensor and each sensor is activated by at most one object. We prove that the impossibility of solving MOTI is a structural problem that depends on the topology of the underlying passage connectivity graph. Then, we describe some restrictions that can be imposed *a priori* either on the graph, on the object movements, or on both, to make MOTI always solvable. More specifically, we show that if the passage connectivity graph is acyclic, MOTI can always be solved. Also, if the graph contains cycles with a length greater than ℓ, MOTI can be solved only if the maximum number of objects that can move concurrently is less than $\lceil \frac{\ell}{2} \rceil$.

Roadmap In Section 2, we discuss some related works. Section 3 introduces the system model. Section 4 defines the Multiple Object Tracking and Identification (MOTI) problem and shows that this problem is impossible to solve in a general setting. Section 5 gives two different characterizations of MOTI solvability, which are used in Section 6 to introduce some interesting classes of systems where the problem can always be solved. Finally, Section 7 concludes the paper.

2 Related Work

Tracking mobile objects through sensors is a problem with a large spectrum of applications [6]. It is treated by the literature from various perspectives, but most of the works are concerned with the problem of correctly tracking one [1] or more [9] objects in a network of binary sensors characterized by noise and false detections under various constraints (*e.g.* noise levels, power consumption [5], limited computational or network resources [7], *etc.*). In our work, we take a more theoretical approach to the problem considering a setting characterized by a network of *ideal* sensors (no noise, no false detection, no limitation on communication) and show that, in the general case,

it is impossible to correctly associate sensed tracks to moving objects. This result is a consequence of the possibility of track merging and splitting during the observation period [7]: once two tracks have merged due to the excessive proximity of two objects, it is impossible to deterministically maintain the identity of the two objects once their tracks split again; To the best of our knowledge, this is the first work to thoroughly examine and discusse this issue as well as prove its strong impact on the general problem solvability.

Issues related to track merging and splitting are common to every setting where two or more objects can move freely. However, in order to provide the reader with proofs of our statements, in this paper, we limit the problem analysis to a specific environment where object movements are partially constrained, and that can be abstracted as a *passage connectivity graph* [8]. Note that, despite the obvious limitations of this model, it still perfectly maps a lot of indoor applications like tracking the movements of people inside a building.

3 System Model

We consider a system composed of a set of generic objects moving in an environment where sensors can detect their presence. Such an environment can be modeled as a passage connectivity graph (PCG) $G(V, E)$ where the set of vertices V represents binary sensors; it exists an edge $e_{i,j} \in E$ linking two vertices $v_i, v_j \in V$ if and only if (*iff*) it is possible for an object to move from the position where it is detected by the sensor v_i to the one where it is detected by the sensor v_j without activating, during the movement, any other sensor. We assume that, in the considered environment, movements are always possible in both directions. Therefore, the graph G is considered undirected. Moreover, we assume that there is always one single way to move from one position to another: only an edge can connect two distinct vertices in the graph. For each vertex v_i, a special edge $e_{i,i}$ exists in E representing the possibility for an object to remain in the same position.

The set of moving objects $\{o_1, \ldots, o_x\}$ is denoted by $\mathbb{O}$. Time is represented as a discrete and infinite sequence $T = [t_0, t_1, \ldots]$ of instants starting at t_0. At every time instant, each object o occupies a position represented by a vertex $v_i \in V$. The position of an object o at a specific time instant t is given by the function $loc : T \times \mathbb{O} \rightarrow V$ that returns the corresponding vertex v_i. When an object, positioned on a vertex v_i, decides to move to a new position, it can choose as its destination any vertex v_j such that $e_{i,j} \in E$; the set of possible destinations is returned by the function $adj : \mathbb{G} \times V \rightarrow \mathcal{P}(V)$. The movement of an object o at a specific time instant t is given by the function $mov : T \times \mathbb{O} \rightarrow E$ that returns the edge used for the movement $e_{i,j}$ where $v_i = loc_t(o)$ and $v_j \in adj_G(v_i)$.

We assume that, regardless of a node's position and movement, the two following conditions always hold in our system: $\forall t \in T, \forall o_i, o_j \in \mathbb{O}$, *(i)* $loc_t(o_i) \neq loc_t(o_j)$ (*i.e.* two objects cannot be located on the same vertex at the same time) and *(ii)* $mov_t(o_i) \neq mov_t(o_j)$ (*i.e.* two objects cannot move on the same edge at the same time).

Each object $o \in \mathbb{O}$ moving in the system describes a trajectory. The *trajectory* described by object o between $t_i, t_j \in T$ (with $j > i$) is defined as $P_{t_i,t_j,o} = [v^{t_i}, \ldots, v^{t_j}]$

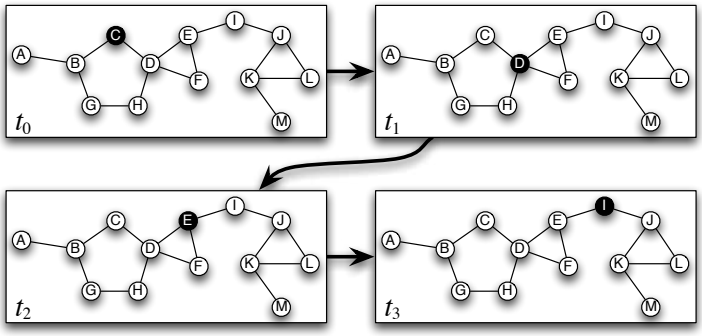

Fig. 1. Example of a trajectory of an object o in the interval $[t_0, t_3]$: $P_{t_0,t_3,o} = [C, D, E, I]$.

where $v^{t_k} = loc_{t_k}(o), k \in [i, j]$. Given t_i and t_j, we define the *global trajectory* as the set containing all object trajectories. The global trajectory is denoted as $P_{t_i,t_j} = \left\{P_{t_i,t_j,o}\right\}_{o\in\mathbb{O}}$, or simply P if the time interval is precisely defined by the context. A specific global trajectory P_{t_i,t_j} is an element of $\mathbb{P}_{t_i,t_j}$, the set containing all possible global trajectories described by the objects in $\mathbb{O}$ on G during the period $[t_i, t_j]$.

3.1 System State

The state of the system at time t is described by the state of each sensor (object detected or not) at that time. This state is represented as a vector of boolean values, one for each vertex in G. In the following, we denote as $\mathbb{S}$ the set of all possible state vectors with respect to x objects in a PCG G. We obviously have $\binom{|V|}{x}$ different state vectors.

Definition 1 (State Vector) *A state vector at time t, denoted S_t, is a vector of size $|V|$ where:*

$$\forall v \in V, S_t[v] = \begin{cases} 1 & \textit{if } \exists\, o \in \mathbb{O} : loc_t(o) = v \\ 0 & \textit{otherwise} \end{cases}$$

3.2 The Observer

The system has an observer that is able to read, at any time t, the state vector S_t. The aim of the observer is to identify objects and trace their trajectories over time. Given a state vector S_t, the observer can use a function tag to assign a unique identifier $\overline{o} \in \overline{\mathbb{O}}$ to each vertex v such that $S_t[v] = 1$ (and a predefined value $\perp$ to all the other vertices), with the only constraint that no two vertices can share a same identifier (with the exception of $\perp$). For instance, if $S_t = [1, 0, 0, 1, 0, 1, 1, 0, 0]$ and $\overline{\mathbb{O}} \subset \mathbb{N}$, we can have $tag(S_t) = [3, \perp, \perp, 2, \perp, 1, 4, \perp, \perp]$. Each identifier represents an object identified by the observer. For the sake of convenience, we introduce the function $\overline{loc} : T \times \overline{\mathbb{O}} \rightarrow V$ that returns the vertex v that was tagged by $\overline{o} \in \overline{\mathbb{O}}$ at time $t \in T$.

Given an object identified by the observer, and its corresponding tag $\overline{o}$, we can define the *observed trajectory* between two time instants t_i, t_j as $\overline{P}_{t_i,t_j,\overline{o}} = [v^{t_i}, \ldots, v^{t_j}]$

where $v^{t_k} = \overline{loc}_{t_k}(\overline{o}), k \in [i,j]$. Effectively, we have to introduce the *observed global trajectory* $\overline{P}_{t_i,t_j} = \left\{\overline{P}_{t_i,t_j,\overline{o}}\right\}_{\overline{o}\in\overline{\mathbb{O}}}$, corresponding to the set of all observed trajectories perceived by the observer. Obviously, $\overline{P}_{t_i,t_j}$ belongs to $\mathbb{P}_{t_i,t_j}$.

4 The Problem of Identifying Objects and Tracking Trajectories

4.1 The MOTI Problem

Let us consider an interval of time $[t_i, t_j]$. The *Multiple Object Tracking and Identification* (MOTI) problem is the problem of defining a function tag such that the following condition holds: $\forall o \in \mathbb{O},\ \exists \overline{o} \in \overline{\mathbb{O}} : P_{t_i,t_j,o} = \overline{P}_{t_i,t_j,\overline{o}}$. In a global view, we can define MOTI as the following condition:

$$P_{t_i,t_j} = \overline{P}_{t_i,t_j} \tag{1}$$

That means that the set of observed trajectories is exactly the same than the real ones. Given that trajectories (both real and observed) are a consequence of object locations, at a finer level of granularity we have:

$$\text{MOTI is solved iff } \forall o \in \mathbb{O},\ \exists \overline{o} \in \overline{\mathbb{O}},\ \forall t \in [t_i, t_j] : \overline{loc_t}(\overline{o}) = loc_t(o)$$

The difficulty of this problem comes from the fact that there could be situations in which the observer can confuse trajectories of two or more objects.

4.2 An Impossibility result

MOTI is not solvable in the cases which are precisely defined by the following theorem:

Theorem 1 (MOTI Unsolvability) *Given an interval of time $[t_i, t_j]$, a PCG $G(V, E)$, a set $\mathbb{O}$ of x objects and a* tag *function, MOTI cannot be solved iff*

$$\exists P, P' \in \mathbb{P}_{t_i,t_j} :\ P \neq P' \wedge \overline{P} = \overline{P'}.$$

where $\overline{P}$ and $\overline{P'}$ are obtained by the observer using the function tag *from real global trajectories P and P'.*

Proof. Let us consider an interval of time $[t_i, t_j]$, a PCG $G(V, E)$ with x objects.

- Assume that (1) $\exists P, P' \in \mathbb{P}_{t_i,t_j} :\ P \neq P' \wedge \overline{P} = \overline{P'}$ and (2) MOTI can be solved. Given that MOTI can be solved we have $P = \overline{P}$ and that $P' = \overline{P'}$ (*cf.* MOTI definition in section 4). But, we have $P \neq P'$ and $\overline{P} = \overline{P'}$. Therefore, there is a contradiction.
- Let prove now that if MOTI cannot be solved $\Longrightarrow \exists P, P' \in \mathbb{P}_{t_i,t_j} :\ P \neq P' \wedge \overline{P} = \overline{P'}$. Let prove the contrapositive (i.e. *modus tollens*) of this proposition.
  Consider $\forall P, P' \in \mathbb{P}_{t_i,t_j} :\ P \neq P' \Longrightarrow \overline{P} \neq \overline{P'}\ \ (P = P' \vee \overline{P} \neq \overline{P'})$.
  Let call $map : \mathbb{P}_{t_i,t_j} \rightarrow \mathbb{P}_{t_i,t_j}$ the function which associate the real trajectory to the observed one, according to the given tag function. Each bijective map function

verifies the following expression: $\forall P, P' \in \mathbb{P}_{t_i,t_j} : P \neq P' \implies map(P) \neq map(P') \implies \overline{P} \neq \overline{P'}$ as $map(\cdot)$ corresponds to the observed trajectories. Let map be the identity function. We have: $\forall P \in \mathbb{P}_{t_i,t_j} : P = map(P) = \overline{P}$. Then, by definition of Section 4, MOTI can be solved. Thus, we have:
MOTI cannot be solved $\implies \exists P, P' \in \mathbb{P}_{t_i,t_j} : P \neq P' \wedge \overline{P} = \overline{P'}$.

We then obtain the equivalence, *i.e.* a characterisation of the MOTI unsolvability. □

Corollary 2 (MOTI Solvability) *Given an interval of time $[t_i, t_j]$, a PCG $G(V, E)$, a set $\mathbb{O}$ of x objects and a* tag *function. MOTI can be solved if, and only if,*

$$\forall P, P' \in \mathbb{P}_{t_i,t_j} : P \neq P' \implies \overline{P} \neq \overline{P'}.$$

Consequently, there is an *impossibility to solve MOTI* if, and only if, there exist at least two global trajectories which respect the condition of Theorem 1. Therefore, we can state that:

Theorem 3 (MOTI Impossibility) *Given the system model presented in Section 3, MOTI is impossible to solve.*

Proof. Consider a 4 vertex PCG G with a 4 edge loop around its vertices, as shown in Figure 2.*a*. Consider two objects moving in this graph G and a time interval constituted by two consecutive time instants $[t_0, t_1]$ with $t_1 = t_0 + 1$. Consider the two global trajectories $P, P' \in \mathbb{P}_{t_0,t_1}$, presented in Figure 2.*b* and 2.*c*. They are defined as:

$$\left\{ \begin{array}{l} P = \left\{ \begin{array}{l} [A\ ,B] \\ [D\ ,C] \end{array} \right\} \\ P' = \left\{ \begin{array}{l} [A\ ,C] \\ [D\ ,B] \end{array} \right\} \end{array} \right.$$

Obviously, we have $P \neq P'$.

According to the system model and the observer capabilities, the information known by the observer for both case P and P', is presented in Figure 2.*d*. We assume that t_0 is the initial time. Then, the observer does not have any other information available about the system than the two following state vectors (which are the same for both P and P'):

$$S_{t_0} = \left[1, 0, 0, 1\right] \text{ and } S_{t_1} = \left[0, 1, 1, 0\right]$$

As the relation tag is a function (a single output for a given input element), it exists a unique tagging for a state, according to the corresponding state vector and previous state available. Thus, assume that:

$$tag(S_{t_0}) = \left[1, \perp, \perp, 2\right].$$

According to $tag(S_{t_0})$ and S_{t_1}, only two values are possible for $tag(S_{t_1})$:

$$(1)\ tag(S_{t_1}) = \left[\perp, 1, 2, \perp\right] \text{ or } (2)\ tag(S_{t_1}) = \left[\perp, 2, 1, \perp\right].$$

For each of these cases, $tag(S_{t_1})$ is the same for the observation of P and P'. So, given a function tag, $\overline{P} = \overline{P'}$. Then, due to Theorem 1, MOTI cannot be solved. □

For better understanding of this result, let's take a closer look at each case.

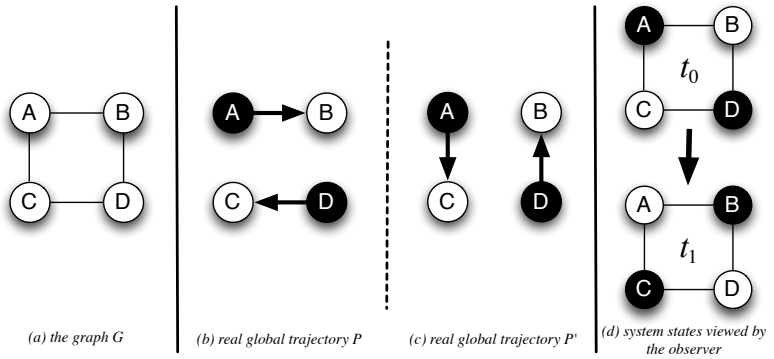

Fig. 2. A simple example in which the observer cannot track precisely the trajectories of objects: For a PCG G presented in (a) with two objects, two global trajectories P and P', presented in (b) and (c) respectively, can occur and the observer is not able to distinguish them with only the information presented on (d) if objects have moved following trajectory P or P'.

Case (1) In this case, the observed global trajectory computed by the observer is:

$$\overline{P} = \overline{P'} = \left\{ \begin{array}{c} [A\ ,B] \\ [D\ ,C] \end{array} \right\}.$$

Then, we have $P = \overline{P} = \overline{P'} \neq P'$ and MOTI is not solvable.

Case (2) By symmetry, in this case, the observed global trajectory computed by the observer is:

$$\overline{P} = \overline{P'} = \left\{ \begin{array}{c} [A\ ,C] \\ [D\ ,B] \end{array} \right\}.$$

Then, we have $P' = \overline{P'} = \overline{P} \neq P$ and MOTI is not solvable.

So, it exists at least one case in which MOTI cannot be solved. Then, Theorem 3 is proved.

5 MOTI Solvability

Before delving into the details about how the system model can be constrained to make MOTI solvable, we need to introduce some other notations.

5.1 Safe and Unsafe Characteristics

Given an object o and the trajectory it describes on the graph as time passes by, we can identify its single movements.

Definition 2 (Movement) *Let $o \in \mathbb{O}$ be an object moving in the system represented by $G(V, E)$. For each time instant $t \in T$, we define its movement as $m_{t,o} = P_{t,t+1,o}$.*

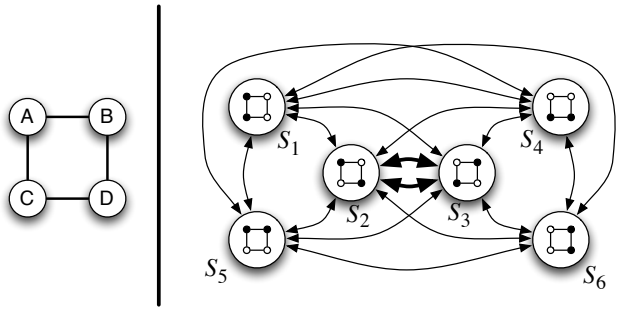

Fig. 3. Example of a state graph for a 4 vertex PCG G (shown on the left) including 2 objects.

Definition 3 (Movement Set) *Consider a system represented by $G(V, E)$ where objects belonging to $\mathbb{O}$ can move. For each time instant $t \in T$, we define the movement set as $M_t = P_{t,t+1}$.*

The movement set is defined, for each time instant t, as the set of all movements done by objects; therefore, it represents how the system "evolves" just after time t.

If we now consider the system state at a specific time t, we can identify all the possible movements that objects are able to do. Each possible combination of these movements corresponds to a different movement set. All these movement sets are defined on the basis of the position of objects on the graph, *i.e.* given a state vector, we can define all the possible movements. From this point of view, movement sets are not necessarily tied to time, as they can be considered as the sets of *possible* movements that objects can do if, at a certain time, they are located on a specific subset of vertices. This idea leads us to the definition of the *State Graph*, which is a graph representing possible system states (in terms of state vectors) and possible movement sets linking them.

Definition 4 (State Graph) *Let $G(V, E)$ be the PCG representing the environment where x objects can move. The corresponding state graph is defined as $SG(\mathbb{S}, \mathbb{M})$, where $\mathbb{S}$ is the set of all the possible state vectors and $\mathbb{M}$ is the set of all the possible movement sets.*

Figure 3 shows an instance of a state graph, depicted on the right side, when considering two moving objects and the 4 vertex graph introduced above, depicted on the left side. Now we can define which edges and which vertices of the state graph should be considered as *unsafe* with respect to the solvability of MOTI.

Definition 5 (Unsafe Movement) *Consider a system represented by $G(V, E)$ where objects belonging to $\mathbb{O}$ can move, and the corresponding state graph $SG(\mathbb{S}, \mathbb{M})$. Consider two states $S, S' \in \mathbb{S}$ such that it exists a movement set $M \in \mathbb{M}$ that links these two states ($S \xrightarrow{M} S'$).*
M is unsafe *iff $\exists M' \in \mathbb{M}$ such that $S \xrightarrow{M'} S'$ and $M \neq M'$.*

We consider unsafe all movement sets linking two system states that are yet linked by some other movement sets. The idea behind this definition is that an observer can not distinguish which movement set has really occurred between all the unsafe movement sets linking the same system states (because the trajectories it observes are calculated only considering system states). The presence of unsafe movements in a state graph make MOTI problem impossible to solve. In the same way, we can define what an unsafe state is.

Definition 6 (Unsafe state) *Consider a system represented by $G(V, E)$ where objects belonging to $\mathbb{O}$ can move, and the corresponding state graph $SG(\mathbb{S}, \mathbb{M})$.*
A state $S \in \mathbb{S}$ is unsafe *iff $\exists M \in \mathbb{M}, \exists S' \in \mathbb{S}$ such that $S \xrightarrow{M} S'$ and M is unsafe.*

Considering again the state graph depicted in Figure 3, states S_2 and S_3 are both unsafe because there are two distinct edges (two unsafe movement sets) linking them (bold arrows). All the other states (and so, all the other movement sets) are safe.

5.2 Characterizing MOTI Solvability

On the basis of these definitions, we revise Theorem 2 and propose two different definitions of MOTI solvability. In the first case, we assume that the global trajectory is known (*e.g.* when analyzing the behaviour of the system *a posteriori*).

Theorem 4 (P-solvability) *Let $SG(\mathbb{S}, \mathbb{M})$ be the state graph defined over a PCG $G(V, E)$ representing the system where objects in $\mathbb{O}$ can move. Consider a specific trajectory $P \in \mathbb{P}_{t_i,t_j}$: MOTI can be solved iff $\forall t \in [t_i, t_{j-1}],\ M_t$ is safe.*

Proof. Consider a specific trajectory $P \in \mathbb{P}_{t_i,t_j}$.

- Assume that $\exists t \in [t_i, t_{j-1}]$ such that M_t is *unsafe*. Let call $P^* = P_{t,t+1}$ the sub-trajectory of the considered P_{t_i,t_j} such that $M_t = P^*$. Due to Definition 5, it exists at least one different unsafe movement M' in $\mathbb{M}$ between the two same system states S, S' such that $M' = P'^*$. Obviously, $P^* \neq P'^*$ but they share the same initial and final state vectors S_t and S_{t+1}. Given that the relation tag available to the observer is a function, it exists a unique tagging for these states. Then, we have:

$$\exists P^*, P'^* \in \mathbb{P}_{t,t+1} : P^* \neq P'^* \wedge \overline{P^*} = \overline{P'^*}.$$

  Therefore, due to Theorem 1, MOTI is unsolvable for P^* and P'^*. Therefore, given that P^* is a sub-trajectory of P_{t_i,t_j}, MOTI is unsolvable for P_{t_i,t_j}.
- Assume here that $\forall t \in [t_i, t_{j-1}],\ M_t$ is *safe*. Then, we have: $\forall t \in [t_i, t_{j-1}]$, $\nexists\, M \neq M_t$ such that $S_t \xrightarrow{M} S_{t+1}$. Then, $\forall t \in [t_i, t_{j-1}], \exists! P \in \mathbb{P}_{t,t+1}$ from S_t to S_{t+1} and, as it is unique, this P is the sub-trajectory between time t and $t+1$ of the considered P_{t_i,t_j}. So, $\forall P' \in \mathbb{P}_{t,t+1}$ such that $P \neq P'$, the initial (or respectively the final) system state of P' is not equal to the initial (or respectively the final) system state of P. Then, consider a bijection function map as introduced in the proof of Theorem 1. We have: $\forall P' \in \mathbb{P}_{t,t+1} : P \neq P' \Longrightarrow map(P) \neq map(P')$. If map is the identity function, we have: $\forall P' \in \mathbb{P}_{t,t+1} : P \neq P' \Longrightarrow \overline{P} \neq \overline{P'}$. Therefore, due to Corollary 2, MOTI can be solved. □

Even though characterizing MOTI solvability with respect to a specific global trajectory is useful for all those systems where we want to decide at any point of time if MOTI is solvable or not, it is also possible to define a set of cases where MOTI is always solvable. This new characterization generalizes Theorem 4 with all possible trajectories that can occur in the system:

Theorem 5 ($\mathbb{P}$-solvability) *Let $SG(\mathbb{S}, \mathbb{M})$ be the state graph defined over a PCG $G(V, E)$ representing the system where objects in $\mathbb{O}$ can move.*
$\forall P \in \mathbb{P}_{t_i,t_j}$: MOTI can be solved iff $\forall S \in \mathbb{S},\ S$ is safe.

Proof. Assume that $\forall P \in \mathbb{P}_{t_i,t_j}$ MOTI can be solved. Then, $\forall P \in \mathbb{P}_{t_i,t_j}, \forall t \in [t_i, t_{j-1}]$: M_t is safe (Theorem 4) and $\forall P \in \mathbb{P}_{t_i,t_j}, \forall t \in [t_i, t_{j-1}]$: S_t is safe (Definition 6). Given that each system state can occur as an initial state of a trajectory, we have:

$$\forall S \in \mathbb{S}, \exists P \in \mathbb{P}_{t_i,t_j} : S_{t_i} = S.$$

Then, given that all system states are safe for all possible trajectories, we have that $\forall S \in \mathbb{S},\ S$ is safe.

Assume now that $\forall S \in \mathbb{S},\ S$ is safe. Due to Definition 6, we have that $\forall M \in \mathbb{M}, M$ is safe. It follows that $\forall P \in \mathbb{P}_{t_i,t_j}, \forall t \in [t_i, t_{j-1}], M$ is safe and, due to Theorem 4 speaks that $\forall P \in \mathbb{P}_{t_i,t_j}$, MOTI can be solved. □

6 A Sufficient Condition for Making MOTI $\mathbb{P}$-solvable

In this section, we show how MOTI can be solved by constraining some characteristics of the system model. More specifically, we show how the problem becomes easily solvable if we assume that object movements are limited in some way.

As we previously explained, the state graph associated to a system may present one or more unsafe states that make MOTI unsolvable. One way to avoid the presence of unsafe states in the state graph, is to remove some of the unsafe movement sets such that the remaining movement sets are all safe. From a practical point of view, this means limiting object movements in the environment, modifying the environment itself or constraining their actions. Let think about a building where people can move from one room to another through doors, this can be realized simply by locking some doors. Therefore, that means deleting the corresponding edges between two vertices of PCG G and modifying accordingly the state graph.

Basically, the only cause of the MOTI non solvability in a system is the presence of cycles in the graph: when two or more objects move inside a cycle, it might be impossible to distinguish their trajectories based on the sole observation of the state vectors. This problem can be avoided by limiting the number of objects in $\mathbb{O}$, which can move concurrently, to a specific k that strictly depends from some characteristics of the PCG G. In this case, we can guarantee that MOTI is solvable for all the possible trajectories P as long as at most k objects move concurrently at each time unit. The set of all these trajectories is a subset of $\mathbb{P}$ and will be denoted as $\mathbb{P}^k$.

Theorem 6 *Let $k \leq |\mathbb{O}|$ be the maximum number of objects that can move concurrently, $\ell > 1$ be an integer and $G(V, E)$ be a PCG which does not contain cycles of length $1 < l < \ell$. $\forall P \in \mathbb{P}^k$: MOTI is P-solvable iff $\left\lceil \frac{\ell}{2} \right\rceil > k$.*

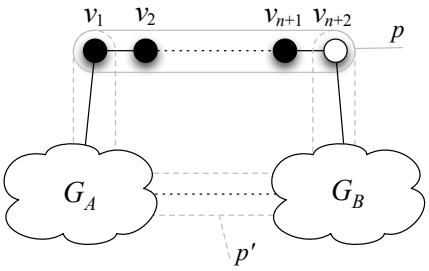

Fig. 4. Representation of the PCG G which illustrates the possibility of exiting paths between v_1 and v_{n+2}.

Proof. We first prove that if $k < \lceil \frac{\ell}{2} \rceil \Rightarrow \forall P \in \mathbb{P}^k$, P does not contain any unsafe movement. The proof is done by induction on the number $n < k$ of objects moving concurrently in the system at each step. In the following, given S and S' two state vectors, we refer to $\mathit{diff}(S, S')$ to denote the number of vertices in G whose state changes between S and S'. Note that $\mathit{diff}(S, S')$ is always an even number because for each object that moves, two vertices change their state in the state vector.

Base step on n. Consider the State Graph construction algorithm reported in Appendix A. At the first iterative step of that algorithm, edges labelled with the movement of a single object are added to the edge-free state graph, linking all the possible couples of state vectors S, S' such that $\mathit{diff}(S, S') = 2$ and it exists an edge in G linking the two vertices that changed their state. For each of these couples (S, S'), a single edge is added as only one object in the system can do the movement associated to the two vertices whose state changes between S and S'. Therefore, for a system where only one object at a time can move (*i.e.* $n = 1$), the resulting state graph does not contain any unsafe movements.

Induction hypothesis. Assume that if $n < k$ objects move concurrently, none of the possible trajectories $P \in \mathbb{P}^n$ contains an unsafe movement.

Induction step on n. Now, consider the case where $n + 1$ objects move concurrently. We want to show that, beside this change, no unsafe movement is added to SG. More specifically, we want to prove that $\forall S, S' \in SG$ such that $2 \leq \mathit{diff}(S, S') \leq 2(n + 1)$, if we can add an edge in SG between S and S' labelled with $n + 1$ concurrent movements, then there cannot exists another edge between them labelled with $n + 1$ or less movements.

Let us first consider the case where $\mathit{diff}(S, S') = 2$. Assume that the two vertices changing their state between S and S' are labelled v_1 and v_{n+2}. An edge labelled with $n + 1$ movements can be added in SG between S and S' only if it the path $p = v_1, v_2, \ldots, v_{n+1}, v_{n+2}$ exists in G and $\forall v \in \{v_1, \ldots, v_{n+1}\}, S[v] = 1$. Now, we show that the only possible movement bringing the system from S to S' is the one where each object located in v_i, with $i \in [1, n + 1]$, moves to v_{i+1}. Assume, without loss of generality, that there is another possible movement that does not involve objects located on vertices $v_2, \ldots, v_{n+1}$. Figure 4 represents this case where the object located

on v_1 must move to a different subgraph G_A of G and an object located on subgraph G_B must move to node v_{n+2}. If the only path connecting vertices in G_A with vertices in G_B is p (it does not exist a link p' as in Figure 4), then $|\{v \in G_A : S'[v] = 1\}| > |\{v \in G_A : S[v] = 1\}|$ (and respectively, $|\{v \in G_B : S'[v] = 1\}| < |\{v \in G_B : S[v] = 1\}|$). This imples that $\exists v \in G_A \cup G_B : S[v] \neq S'[v] \Rightarrow$*diff*$(S, S') > 2$, which is impossible due to our initial hypothesis. On the contrary, if there is a path $p' \neq p$ connecting G_A to G_B, then p is part of a cycle $c \subset G$. The length of c is, by assumption, at least ℓ. In order to have *diff*$(S, S') = 2$, in S, there must be an object located on all vertices in c but v_{n+2} and all the objects located on these vertices, with the exception of those located on $v_2, \ldots, v_{n+1}$, must be moved (otherwise *diff*$(S, S') > 2$). But this means that m ($\geq \ell - n$) objects will move concurrently. Given that $\lceil \frac{\ell}{2} \rceil > k \geq n+1$, we have $m > 2(n+1) - n > n+1$, *i.e.* every other edges connecting S to S' in SG must be labelled with more than $n+1$ movements.

Now, consider the case where *diff*$(S, S') = 2(x+1)$ with $x \leq n$. In this case, there are $n+1$ distinct objects moving on $x+1$ distinct paths, each characterized by the presence of an object on each vertex but the last one (as p in Figure 4). The same reasoning shown for the previous case can be applied to every single path, considering that each of these paths contains strictly less than $n+1$ objects.

Now, we prove by contradiction that if $\forall P \in \mathbb{P}^k$, MOTI is P-solvable $\Rightarrow \lceil \frac{\ell}{2} \rceil > k$. Assume for the moment that $k \geq \lceil \frac{\ell}{2} \rceil$. Consider the smallest cycle c in G constituted by vertices $v_1, \ldots, v_\ell$. Now consider, without loss of generality, a state vector S where $\forall i \in \{2 \cdot k - 1 | k \in [1, \lceil \frac{\ell}{2} \rceil]\} : S[v_i] = 1$ and $\forall i \in \{2 \cdot k | k \in [1, \lfloor \frac{\ell}{2} \rfloor]\} : S[v_i] = 0$. Consider also the state vector S' that is identical to S but where $\forall i \in \{2 \cdot k - 1 | k \in [1, \lfloor \frac{\ell}{2} \rfloor]\} : S[v_i] = 0$ and $\forall i \in \{2 \cdot k | k \in [1, \lfloor \frac{\ell}{2} \rfloor]\} : S[v_i] = 1$ (Note that, if ℓ is odd then $S'[v_\ell]$ remains unchanged and equals to 1). Then, the states of all vertices – but the last in case of ℓ odd – indexed from 1 to ℓ are inverted. Such two state vectors are certainly in SG because we are assuming that $k \geq \lceil \frac{\ell}{2} \rceil$. Now consider the following movements $M = \{[v_i, v_{i+1 \mod \ell}\}_{i \in \{2 \cdot k - 1 | k \in [1, \lfloor \frac{\ell}{2} \rfloor]\}}$ and $M' = \{[v_i, v_{i-1 \mod \ell}\}_{i \in \{2 \cdot k - 1 | k \in [1, \lceil \frac{\ell}{2} \rceil]\}}$. Both movements link S to S' in SG and are labeled with lower or equals concurrent moves than k. Therefore, it exists unsafe movements. This is in contradiction with the initial assumption that $\forall P \in \mathbb{P}^k$, MOTI is P-solvable. □

In a practical setting, there are two possible methods to guarantee that the conditions at the basis of Theorem 6 always hold. The first method requires to choose, as the PCG, a topology characterized by cycles of length strictly larger than $2 \cdot x$. As a consequence, MOTI is $\mathbb{P}$-solvable in any system characterized by an acyclic graph as presented in the following Theorem 7. The second method requires to limit the number of objects that can move concurrently in the system.

Theorem 7 *MOTI is $\mathbb{P}$-solvable in any system characterized by an acyclic graph $G(V, E)$.*

Proof. We can consider G as a graph with a cycle of infinite length. Due to Theorem 6, MOTI is $\mathbb{P}$-solvable as long as no more than an infinite number of objects move concurrently in the system. Given that $\mathbb{O}$ is finite, the $\mathbb{P}$-solvability is always guaranteed. □

7 Conclusion

In this paper, we have considered the possibility of assigning trajectories to moving objects on a generic sparse passage connectivity graph. We have shown that assigning a unique identifier to a track revealed by a binary sensor network, namely the MOTI problem, is impossible. To make the result as strong as possible, it has been proved under strong assumptions i.e., perfect binary sensors, perfect coverage and omniscient observer of the state of the graph. We proved that this impossibility depends primarily of the topology of the graph and secondly of the number of moving objects.

Following this impossibility result, we have considered restricting the model to make such assignments possible. More specifically, we have shown that MOTI can be solved either if the graph is acyclic or the length of the smallest cycle in the graph is strictly greater than the double of the number of objects that can concurrently move in the system. This leaves the opportunity to modify the topology of the passage connectivity graph like for example in an indoor scenario. Once the maximum number of moving objects is known, our results mainly impact on the deployment phase of a sensor network.

Lot of open questions have still to be considered. For instance, some characterizations can be reformulated in order to reduce the computation of the observer. In a distributed way, identifying distributed algorithms able to impose the aforementioned restrictions to the system model, without relying on a global knowledge of the system is not trivial. This leaves the space for an exciting research agenda.

References

1. Javed Aslam, Zack Butler, Florin Constantin, Valentino Crespi, George Cybenko, and Daniela Rus. Tracking a moving object with a binary sensor network. In *Proc. of SenSys'03*, 2003.
2. James Aspnes, David Goldengerg, and Yang Richard Yang. On the computational complexity of sensor network localization. In *Proc. of 1st Intl Workshop on Algorithmic Aspects of WSN*, 2004.
3. Domenico Bloisi, Luca Iocchi, G. R. Leone, R. Pigliacampo, L. Tombolini, and Lorena Novelli. A distributed vision system for boat traffic monitoring in the venice grand canal. In *VISAPP'07*, 2007.
4. Robert W. Floyd. Algorithm 97: Shortest path. *Comm. of the ACM*, 1962.
5. Chao Gui and Prasant Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *Proc. of MobiCom'04*, 2004.
6. Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An algorithm for multiple object trajectory tracking. In *Proc. of CVPR'04*, 2004.
7. Juan Liu, Jie Liu, James Reich, Patrick Cheung, and Feng Zhao. Distributed group management for track initiation and maintenance in target localization applications. In *Proc. of IPSN'03*, 2003.
8. Songhwai Oh and Shankar Sastry. Tracking on a graph. In *Proc. of IPSN '05*, 2005.
9. Jaspreet Singh, Upamanyu Madhow, Rajesh Kumar, Subhash Suri, and Richard Cagley. Tracking multiple targets using binary proximity sensors. In *Proc. of IPSN'07*, 2007.
10. Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM*, 1962.

A An Algorithm for Computing SG

In this appendix, we propose a state graph construction algorithm, based on the conditional transitive closure of a one-movement only state graph. This condition ensures that no inconsistent movement will be included in the state graph generated by the algorithm.

This protocol is composed of two main parts. The first part creates an empty edge set and a complete vertices set. Then, starting from line 3 to line 6, the algorithm puts in the state graph all one-object-movement edges. From line 7 to line 13, following the same mechanism as the one used in the Floyd-Warshall algorithm for transitive closure computation, three nested loops compute iteratively all possible movements with any number of concurrent movements. Finally, the last lines from 14 to 15, merge symmetric movement edges in order to return an undirected state graph.

The termination of this protocol is trivial as it is composed only of nested loops of finite length. Moreover, the complexity of this algorithm is obviously $O(|\mathbb{S}|^3)$ as it contains three nested loops on $\mathbb{S}$.

The correctness of this algorithm is based on the correctness of the Floyd-Warshall algorithm [4, 10]. The condition of line 12 only ensures that no inconsistent movement can be included in the state graph, and given that new edges in SG are generated from consistent movements, no existing movement can be ignored.

Algorithm 1: State graph construction

Data: a graph G, the number of moving object x
Result: the associated state graph $SG(\mathbb{S}, \mathbb{M})$

1 $\mathbb{S} \leftarrow \{S \in \{0,1\}^{|V|} | \sum_{v \in V} S[v] = x\}$;
2 $\mathbb{M} \leftarrow \varnothing$;
3 **foreach** $S \in \mathbb{S}$ **do**
4 **foreach** $S' \in \mathbb{S} \backslash \{S\}$ **do**
5 **if** $\exists!(v, v') \in V2$ *such that* $S[v] = S'[v'] = 1$ *and* $S[v'] = S'[v] = 0$ **then**
6 $\mathbb{M} \leftarrow \mathbb{M} \cup \{S \xrightarrow{\{[v-v']\}} S'\}$;
7 **foreach** $S \in \mathbb{S}$ **do**
8 **foreach** $S' \in \mathbb{S} \backslash \{S\}$ **do**
9 **foreach** $S'' \in \mathbb{S} \backslash \{S\}$ **do**
10 **if** $(S' \rightarrow S \in \mathbb{M}) \wedge (S \rightarrow S'' \in \mathbb{M})$
11 **and** $(\forall S' \rightarrow S'' \in \mathbb{M},\ e_{S',S''} \neq e_{S',S} \cup e_{S,S''})$
12 **and** $\forall [v', v_1] \in e_{S',S},\ \forall [v_2, v''] \in e_{S,S''},\ v_1 \neq v_2$ **then**
13 $\mathbb{M} \leftarrow \mathbb{M} \cup \{S' \xrightarrow{e_{S',S} \cup e_{S,S''}} S''\}$;
14 **if** G *is undirected* **then**
15 merge symetric edges in $\mathbb{M}$;
16 **return** $(\mathbb{S}, \mathbb{M})$;