Theoretical Computer Science ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the exploration of time-varying networks[☆]

Paola Flocchini[a], Bernard Mans[b,*], Nicola Santoro[c]

[a] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
[b] Department of Computing, Macquarie University, Sydney, Australia
[c] School of Computer Science, Carleton University, Ottawa, Canada

ARTICLE INFO

Article history:
Received 17 March 2011
Received in revised form 4 December 2011
Accepted 23 October 2012
Communicated by R. Klasing

Keywords:
Time-varying graphs
Exploration
Dynamic networks
Carrier networks
Evolving graphs
Traversal
Mobile networks
Mobile agents
Delay-tolerant networks

ABSTRACT

We study the computability and complexity of the *exploration problem* in a class of highly dynamic networks: *carrier* graphs, where the edges between sites exist only at some (unknown) times defined by the periodic movements of mobile carriers among the sites. These graphs naturally model highly dynamic infrastructure-less networks such as public transports with fixed timetables, low earth orbiting (LEO) satellite systems, security guards' tours, etc. We focus on the *opportunistic* exploration of these graphs, that is by an agent that exploits the movements of the carriers to move in the network.

We establish necessary conditions for the problem to be solved. We also derive lower bounds on the amount of time required in general, as well as for the carrier graphs defined by restricted classes of carrier movements.

We then prove that the limitations on computability and complexity we have established are indeed tight. In fact we prove that all necessary conditions are also sufficient and all lower bounds on costs are tight. We do so constructively by presenting two optimal solution algorithms, one for anonymous systems, and one for those with distinct node IDs.

Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.

1. Introduction

1.1. The framework

Graph exploration is a classical fundamental problem extensively studied since its initial formulation in 1951 by Shannon [32]. It has various applications in different areas, e.g., finding a path through a maze, or searching a computer network using a mobile software agent. In these cases, the environment to be explored is usually modelled as a (di)graph, where a single entity (called agent or robot) starting at a node of the graph, has to visit all the nodes and terminate within finite time. Different instances of the problem exist depending on a variety of factors, including whether the nodes of the graph are labelled with unique identifiers or are anonymous, the amount of memory with which the exploring agent is endowed, the amount of a priori knowledge available about the structure of the graph (e.g., it is acyclic) etc. (e.g., see [1,3,6,15,16,18]). In spite of their differences, all these investigations have something in common: they all assume that the graph to be explored is *connected*.

The connectivity assumption unfortunately does not hold for the new generation of networked environments that are highly dynamic and evolving in time. In these infrastructure-less networks, end-to-end multi-hop paths may not exist, and

☆ This work was partially supported by ARC, ANR Project SHAMAN, by COST Action 295 DYNAMO, and by NSERC. A preliminary and shorter version of this work was presented at the 20th International Symposium on Algorithms and Computation.

* Corresponding author. Tel.: +61 2 98509574; fax: +61 2 98509551.
E-mail addresses: flocchin@eecs.uottawa.ca (P. Flocchini), bernard.mans@mq.edu.au (B. Mans), santoro@scs.carleton.ca (N. Santoro).

0304-3975/$ – see front matter Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.10.029

it is actually possible that, at every instant of time, the network is disconnected. However, communication routes may be available through time and mobility, and not only basic tasks like routing, but complex communication and computation services could still be performed. See in this regard the ample literature (mostly from the application/engineering community) on these highly dynamic systems, variously called *delay tolerant*, *disruption tolerant*, *challenged*, and *opportunistic* networks (e.g., [5,8,9,14,19,22,24,25,29–31,33,36,37]). Almost all the existing work in this area focuses on the *routing* and the *broadcast* problems. In these cases, most recent results consider these problems from a probabilistic standpoint: e.g., in [5,14], where the changes in the graph are regulated by a Markovian process on the edges. Deterministic solutions to routing and broadcasting have also been obtained under strong assumptions such as knowing the complete edge schedule ahead of time, and within a centralized entity (see [8]); intermediate assumptions have also been investigated, such as the fact that the network is always connected [31].

The highly dynamic features of dynamic networks can be described by means of *time-varying* graphs, that is graphs where links between nodes exist only at some times (a priori unknown to the algorithm designer). Thus, in these graphs, (also called *evolving* graphs or *temporal* graphs), the set of edges existing at a given time might not form a connected graph (e.g., see [10–13,17,19,26,28,34]).

In spite of the large amount of the literature and research investigations, no work exists on *exploration* of such networks, with the noticeable exception of the study of exploration by random walks in dynamic undirected graphs [2]. In that paper, it is shown that unfortunately, and contrarily to results for connected static undirected graphs, when the graph is modified by an oblivious adversary, this adversary can devise strategies which force the expected cover time of a simple random walk to be exponential. However the authors prove that a simple max-degree random walk guarantees expected polynomial cover time regardless of the change made by the adversary, and thus confirm the exploitability of dynamic networks.

Our research interest is on the deterministic exploration of time-varying graphs, on the computability and complexity aspects of this problem.

1.2. The problem

In this paper, we start the investigation focusing on a particular class of time-varying graphs: the *carrier graphs* (C-graphs), where the edges of the graphs are defined by the periodic movements of some mobile entities, called carriers. This class models naturally infrastructure-less networks where mobile entities have fixed routes that they traverse regularly. Examples of such common settings are public transports with fixed timetables, low earth orbiting (LEO) satellite systems, security guards' tours, etc. These networks have been investigated in the application/engineering community, with respect to routing and to the design of carriers' routes (e.g., see [22,30]) and more specifically for buses [4,36], planes [27] and satellites [35]. The use of such carrier networks allows the creation of a powerful routing system based upon regular contacts. This in turn allows the development of a wide range of applications, which vary from simply expanding content delivery through the opportunistic use of computing devices on common carriers (such as buses, trains, planes, etc.), to providing a fully reliable interplanetary overlay network that bridges smaller local (space) networks automatically without having to manually schedule dedicated time for routing data [35].

At the basis is the fact that messages, code, and content can travel on a carrier, and can be transmitted to another carrier when the two meet (i.e., they are in communication range). This situation can be abstracted in terms of an agent opportunistically transported by a carrier, and moving from that carrier to an encountered carrier. Our interest is in understanding how exploration of C-graphs can take place, performed by such an opportunistic agent; in particular we focus on how the periodicity of the carrier routes impacts the complexity of the problem.

We view the system as composed of n sites (or nodes) and k carriers, each periodically moving among a subset of the sites. The routes of the carriers define the edges of the time-varying graph: a directed edge (or arc) exists from node u to node v at time t only if there is a carrier that in its route moves from u to v at time t. If all routes have the same period p the system is called *homogeneous*, otherwise it is called *heterogeneous* (and p is the length of the longest route).

In the system enters an explorer agent **A** that can ride with any carrier along its route, and it can switch to any carrier it meets while riding. We do not assume that the agent can disembark and wait for a carrier at the site. Such an assumption would require the sites to have storage capabilities (to hold the agent), and our focus is on solving the problem under minimal capabilities. Although for some specific applications it may be possible for agents to wait at a site in order to jump on another carrier later (e.g., agent associated to human-carried devices in a bus network), in other applications this assumption does not apply. As an example, consider a static sensor network deployed over a terrain (e.g., at bus stops) to collect local data (e.g., local traffic); carriers (e.g., buses) endowed with wireless communication capabilities move along the sensors' locations; a data gathering agent rides on the carriers, when at a sensor location, it collects the sensor's data, and when encountering another carrier it can migrate there; the goal is for the agent to collect the data from all sensors.

Exploring a C-graph is the process of **A** visiting all the nodes and exiting the system within finite time. We study the computability and the complexity of the exploration problem of C-graphs, *CG-Exploration*.

1.3. Overview of results

We first investigate the computability of *CG-Exploration* and establish necessary conditions for the problem to be solvable. We prove that in *anonymous* systems (i.e., the nodes have no identifiers) exploration is unsolvable if the agent has no knowledge of (an upper bound on) the size of the largest route; if the nodes have *distinct IDs*, we show that either n or an upper-bound on the system period must be known for the problem to be solvable.

These necessary conditions for anonymous systems, summarized in the table below (see Fig. 1), hold even if the routes are *homogeneous* (i.e., have the same length), the agent has unlimited memory and knows k (if anonymous, even if they know n).

ANONYMOUS		
Knowledge	Solution	(Even if)
(bound on) p unknown	*impossible*	n, k known; homogeneous unbounded memory
(bound on) p known	*possible*	n, k unknown; heterogeneous $O(\log p + k \log k)$ bits

DISTINCT IDs		
Knowledge	Solution	(Even if)
n and (bound on) p unknown	*impossible*	k known; homogeneous unbounded memory
n known	*possible*	p, k unknown; heterogeneous $O(n \log n)$ bits
(bound on) p known	*possible*	n, k unknown; heterogeneous $O(\log p + k \log k)$ bits

Fig. 1. Results for the Computability of CG-Explorations.

We then consider the complexity of *CG-Exploration* and establish lower bounds on the number of moves. We prove that in general $\Omega(kp)$ moves are necessary for homogeneous systems and $\Omega(kp^2)$ for heterogeneous ones. This lower bound holds even if **A** knows n, k, p, and has unlimited memory. Notice that the parameter p in the above lower bounds can be arbitrarily large since the same node can appear in a route arbitrarily many times. A natural question is whether the lower bounds do change when imposing restrictions on the "redundancy" of the routes. To investigate the impact of the routes' structure on the complexity of the problem, we consider C-graphs where all the routes are *simple*, that is, do not contain self-loops or multiple arcs. We show that the same type of lower bound holds also for this class; in fact, we establish $\Omega(kn^2)$ lower bound for homogeneous and $\Omega(kn^4)$ lower bound for heterogeneous systems with simple routes. We then further restrict each route to be *circular*, that is an arc appears in a route at most once. Even in this case, the general lower bound holds; in fact, we prove lower bounds of $\Omega(kn)$ moves for homogeneous and $\Omega(kn^2)$ for heterogeneous systems with circular routes. Interestingly these lower bounds hold even if **A** has full knowledge of the entire C-graph, and has unlimited memory. We then prove that the limitations on computability and complexity established so far, are indeed tight. In fact, we prove that all necessary conditions are also sufficient and all lower bounds on costs are tight. We do so by constructively presenting two worst case optimal solution algorithms, one for anonymous systems and one for those with IDs. In the case of *anonymous* systems, the algorithm solves the problem without requiring any knowledge of n or k; in fact, it only uses the necessary knowledge of an upper bound $B \geq p$ on the size of the longest route. The number of moves is $O(kB)$ for homogeneous and $O(kB^2)$ for heterogeneous systems. The cost depends on the accuracy of the upperbound B on p. It is sufficient that the upper bound B is linear in p for the algorithm to be *optimal*. In the case of systems *with IDs*, the algorithm solves the problem without requiring any knowledge of p or k; in fact it only uses the necessary knowledge of n. The number of moves is $O(kp)$ and $O(kp^2)$ matching the lower bound (see Fig. 2).

	System	
Routes	*Homogeneous*	*Heterogeneous*
Arbitrary	$\Theta(kp)$	$\Theta(kp^2)$
Simple	$\Theta(kn^2)$	$\Theta(kn^4)$
Circular	$\Theta(kn)$	$\Theta(kn^2)$

Fig. 2. Results for the complexity of CG-explorations.

Our optimal solutions are further enhanced by their simplicity and by the use of a limited amount of memory.

2. Model and terminology

2.1. Carrier graphs

The system is composed of a set S of n *sites*; depending on whether the sites have unique IDs or no identifiers, the system will be said to be *with IDs* or *anonymous*, respectively. In the system operates a set C of k mobile entities called *carriers* moving among the sites. Considering the low density of traffic in applications for Delay-Tolerant Networks, we will assume, w.l.o.g. that $|C| = k \leq n = |S|$. Each carrier c has a unique identifier $id(c)$ and an ordered sequence of sites $\pi(c) = \langle x_0, x_1, \ldots, x_{p(c)-1}\rangle$, $x_i \in S$, called *route*; for any integer j we will denote by $\pi(c)[j]$ the component x_i of the route where $i \equiv j \mod p(c)$, and $p(c)$ will be called the *period* of $\pi(c)$. A carrier $c \in C$ moves cyclically along its *route* $\pi(c)$: at time t, c will move from $\pi(c)[t]$ to $\pi(c)[t+1]$. In the following, x_0 will be called the *starting* site of c, and the set $S(c) = \{x_0, x_1, \ldots, x_{p(c)-1}\}$, will be called the *domain* of c; clearly $|S(c)| \leq p(c)$ (as the same site can be visited several times along the route).

Each route $\pi(c) =< x_0, x_1, \ldots, x_{p(c)-1} >$ defines an arc-labelled multigraph $\vec{G}(c) = (S(c), \vec{E}(c))$, where $\vec{E}(c) = \{(x_i, x_{i+1}, i), 0 \leq i < p(c)\}$, and the operations on the indices are modulo $p(c)$. If $(x, y, t \mod p(c)) \in \vec{E}(c)$, we shall say that c *activates* the arc (x, y) at time t. A site $z \in S$ is the *meeting point* (or *connection*) of carriers a and b at time t if $\pi(a)[t] = \pi(b)[t] = z$; that is, there exist sites x and y such that, at time $t-1$, a activates the arc (x, z) and b activates the arc (y, z). A route $\pi(c) =< x_0, x_1, \ldots, x_{p(c)-1} >$ is *simple* if $\vec{G}(c)$ does not contain self loops or multiple arcs; that is $x_i \neq x_{i+1}$, for $0 \leq i < p(c)$, and if $(x, y, i), (x, y, j) \in \vec{E}(c)$ then $i = j$.

A simple route $\pi(c)$ is *irredundant* (or *circular*) if $\vec{G}(c)$ is either a simple cycle or a simple traversal with return from a root of a tree: in other words, the undirected graph induced by the simple route is either a cycle or a tree. Note that in this case, $|\vec{E}(c)| = p(c) \leq 2(n-1)$.

We shall denote by $R = \{\pi(c) : c \in C\}$ the set of the routes of all carriers and, as these routes are periodic, by $p(R) = \mathrm{Max}\{p(c) : c \in C\}$ the maximum *period* among all the routes of the carriers. When no ambiguity arises, we will denote $p(R)$ simply as p. The set R defines an arc-labelled multigraph $\vec{G}_R = (S, \vec{E})$, where $\vec{E} = \cup_{c\in C}\vec{E}(c)$, called *carrier graph* (or, shortly, *C-graph*).

A *concrete walk* (or, simply, *walk*) σ in $\vec{G}_R$ is a (possibly infinite) ordered sequence $\sigma =< e_0, e_1, e_2 \ldots >$ of arcs in $\vec{E}$ where $e_i = (a_i, a_{i+1}, i) \in \vec{E}(c_i)$ for some $c_i \in C$, $0 \leq i$. To each route $\pi(c)$ in R corresponds an infinite concrete walk $\sigma(c)$ in $\vec{G}_R$ where $e_i = (\pi(c)[i], \pi(c)[i+1], i)$ for $i \geq 0$. A concrete walk σ is a *concrete cover* of $\vec{G}_R$ if it includes every site: $\cup_{0\leq i\leq|\sigma|+1} \{a_i\} = S$.

A set of routes R is *feasible* if there exists at least one concrete cover of $\vec{G}_R$ starting from any carrier. R is *homogeneous* if all routes have the same period: $\forall a, b \in C,\ p(a) = p(b)$; it is *heterogeneous* otherwise. R is *simple* (resp. *irredundant*) if every route $\pi(c) \in R$ is simple (resp., irredundant). Please note that this is when each route is considered individually: even if all the routes are simple, the resulting system $\vec{G}_R$ is not necessarily simple (resp., irredundant). We will give several examples later. With an abuse of notation, the above properties of R will be used also for $\vec{G}_R$; hence we will accordingly say that $\vec{G}_R$ is feasible (or homogeneous, simple, etc.).

In summary, when no ambiguity arises, we will denote $p(R)$ simply as p, $\vec{G}_R$ simply as $\vec{G}$, and $(x, y, t \mod p(c))$ simply as (x, y, t).

Examples of C-graphs with arbitrary routes, with simple routes and with circular routes are given in Figs. 3, 4 and 6 respectively.

2.2. Exploring agent and traversal

In the system is injected an external computational entity **A** called the *exploring agent*; w.l.o.g., the agent is injected at the starting site of some carrier at time $t = 0$. The only two operations it can perform are: move with a carrier, switch carrier. Agent **A** can switch from carrier c to carrier c' at site y at time t iff it is riding with c at time t and both c and c' arrive at y at time t, that is: iff it is riding with c at time t and $\exists\, x, x' \in S$ such that $(x, y, t) \in E(c)$ and $(x', y, t) \in E(c')$.

Agent **A** does not necessarily know n, k, or $\vec{G}$; when at a site x at time t, **A** can however determine the identifier $id(c)$ of each carrier c that arrives at $x \in S$ at time t.

The goal of **A** is to fully explore the system within finite time, that is to *visit* every site and terminate, exiting the system, within finite time, regardless of the starting position. We will call this problem *CG-Exploration*.

An *exploration protocol* $\mathcal{A}$ is an algorithm that specifies the exploring agent's actions enabling it to traverse C-graphs. More precisely, let $start(\vec{G}_R) = \{\pi(c)[0] : c \in C\}$ be the set of starting sites for a C-graph $\vec{G}_R$, and let $C(t, x) = \{c \in C : \pi(c)[t] = x\}$, be the set of carriers that arrive at $x \in S$ at time $t \geq 0$. Initially, at time $t = 0$, **A** is at a site $x \in start(\vec{G}_R)$. If **A** is at node y at time $t \geq 0$, $\mathcal{A}$ specifies $action \in C(t, x) \cup \{\text{halt}\}$: if $action = c \in C(t, x)$, **A** will move with c to $\pi(c)[t+1]$, traversing the edge $(x, \pi(c)[t+1], t)$; if $action = \text{halt}$, **A** will terminate the execution and exit the system. Hence the *execution* of $\mathcal{A}$ in $\vec{G}_R$ starting from injection site x uniquely defines the (possibly infinite) concrete walk $\xi(x) =< e_0, e_1, e_2, \cdots >$ of the arcs traversed by **A** starting from x; the walk is infinite if **A** never executes $action = \text{halt}$, finite otherwise.

Algorithm $\mathcal{A}$ *solves* the *CG-Exploration* of $\vec{G}_R$ if $\forall x \in start(\vec{G}_R)$, $\xi(x)$ is a finite concrete cover of $\vec{G}_R$; that is, executing $\mathcal{A}$ in $\vec{G}_R$, **A** visits all sites of $\vec{G}_R$ and performs $action = \text{halt}$, regardless of the injection site $x \in start(\vec{G}_R)$. Clearly, we have the following property.

Property 2.1. CG-Exploration *of $\vec{G}_R$ is possible only if R is feasible.*

Hence, in the following, we will assume that R is *feasible* and restrict *CG-Exploration* to the class of feasible C-graphs. We will say that problem *CG-Exploration* is *unsolvable* (in a class of C-graphs) if there is no deterministic exploration algorithm that solves the problem for all feasible C-graphs (in that class).

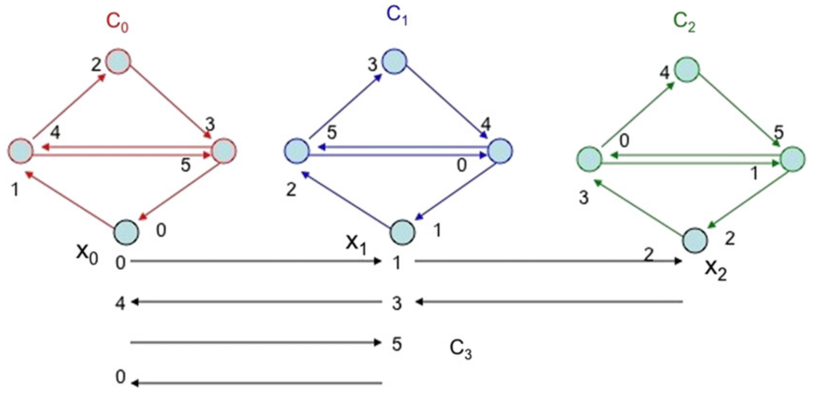

Fig. 3. C-graph of Theorem 3.3 with $n = 12, k = 4, p = 6$.

The cost measure is the number of *moves* that the exploring agent **A** performs. Let $\mathcal{M}(\vec{G}_R)$ denote the number of moves that need to be performed in the worst case by **A** to solve *CG-Exploration* in feasible $\vec{G}_R$. Given a class $\mathcal{G}$ of feasible graphs, let $\mathcal{M}(\mathcal{G})$ be the largest $\mathcal{M}(\vec{G}_R)$ over all $\vec{G}_R \in \mathcal{G}$ and let $\mathcal{M}_{homo}(n, k)$ (resp. $\mathcal{M}_{hetero}(n, k)$) denote the largest $\mathcal{M}(\vec{G}_R)$ in the class of all feasible homogeneous (resp. heterogeneous) C-graphs $\vec{G}_R$ with n sites and k carriers.

3. Computability and lower bounds

3.1. Knowledge and solvability

The availability of a priori knowledge by **A** about the system has an immediate impact on the solvability of the problem *CG-Exploration*. Consider first *anonymous* systems: the sites are indistinguishable to the exploring agent **a.** In this case, the problem is unsolvable if **A** has no knowledge of (an upper bound on) the system period.

Theorem 3.1. *Let the system be* anonymous. CG-Exploration *is unsolvable if* **A** *has no information on (an upper bound on) the system period. This result holds even if the system is restricted to be* homogeneous, **A** *has unlimited memory and knows both n and k.*

Proof. By contradiction, let $\mathcal{A}$ solve *CG-Exploration* in all anonymous feasible C-graphs without any information on (an upper bound on) the system period. Given n and k, let $S = \{x_0, \dots, x_{n-1}\}$ be a set of n anonymous sites, and let π be an arbitrary sequence of elements of S such that all sites are included. Consider the homogeneous system where k carriers have exactly the same route π and let $\vec{G}$ be the corresponding graph. Without loss of generality, let x_0 be the starting site. Consider now the execution of $\mathcal{A}$ by **A** in $\vec{G}$ starting from x_0. Since $\mathcal{A}$ is correct, the walk $\xi(x_0)$ performed by **A** is a finite concrete cover; let m be its length. Furthermore, since all carriers have the same route, $\xi(x_0)$ is a prefix of the infinite walk $\sigma(c)$, performed by each carrier c; more precisely it consists of the first m arcs of $\sigma(c)$. Let t_i denote the first time when x_i is visited in this execution; without loss of generality, let $t_i < t_{i+1}$, $0 \le i < n-2$.

Let π^* denote the sequence of sites in the order they are visited by **A** in the walk $\xi(x_0)$. Let α be the first $t_{n-2} + 1$ sites of π^*, and β be the next $m + 1 - (t_{n-2} + 1)$ sites (recall, m is the length of $\xi(x_0)$ and thus $m + 1$ is that of π^*). Let γ be the sequence obtained from β by substituting each occurrence of x_{n-1} with x_{n-2}.

Consider now the homogeneous system where all the k agents have the same route $\pi' =< \alpha, \gamma, \beta >$, and let $\vec{G}'$ be the corresponding graph.

The execution of $\mathcal{A}$ in $\vec{G}'$ by **A** with injection site x_0 results in **A** performing a concrete walk $\xi'(x_0)$ which, for the first m arcs, is identical to $\xi(x_0)$ except that each arc of the form (x, x_{n-1}, t) and (x_{n-1}, x, t) has been replaced by (x, x_{n-2}, t) and (x_{n-2}, x, t), respectively. Because of anonymity of the nodes, **A** will be unable to distinguish x_{n-1} and x_{n-2}; furthermore, it does not know (an upper bound on) the system's period. Thus **A** will be unable to distinguish the first m steps of the two executions; it will therefore stop after m moves also in $\vec{G}'$. This means that **A** stops before traversing β; since x_{n-1} is neither in α nor in γ, $\xi'(x_0)$ is finite but not a concrete cover of $\vec{G}'$, contradicting the correctness of $\mathcal{A}$. □

In other words, in anonymous systems, an upper bound on the system period must be available to **A** for the problem to be solvable.

Consider now *distinct ID's* systems, i.e. where the sites have distinct identities accessible to **A** when visiting them; in this case, the problem is unsolvable if **A** has no knowledge of (an upper bound on) the system period or of the number of sites.

Theorem 3.2. *Let the sites have* distinct IDs. CG-Exploration *is unsolvable if* **A** *has no information on either (an upper bound on) the system period or of the number of sites. This result holds even if the system is* homogeneous, *and* **A** *has unlimited memory and knows k.*

Proof. By contradiction, let $\mathcal{A}$ solve *CG-Exploration* in all feasible C-graphs with *distinct IDs* without any information on either (an upper bound on) the system period or on the number of sites. Let $S = \{x_0, \ldots, x_{n-1}\}$ be a set of n sites with distinct IDs, and let π be an arbitrary sequence of elements of S such that all sites are included. Consider now the homogeneous system where k carriers have exactly the same route π and let $\vec{G}$ be the corresponding graph. Without loss of generality, let x_0 be the starting site.

Consider now the execution of $\mathcal{A}$ by **A** in $\vec{G}$ starting from x_0. Since $\mathcal{A}$ is correct, the walk $\xi(x_0)$ performed by **A** is a finite concrete cover; let m be its length and let $\overline{\pi}$ be the corresponding sequence of nodes. Furthermore, since all carriers have the same route, $\xi(x_0)$ is a prefix of the infinite walk $\sigma(c)$, performed by each carrier c; more precisely it consists of the first m arcs of $\sigma(c)$. Consider now the homogeneous system with $n+1$ sites, with $S' = \{x_0, \ldots, x_{n-1}, x_n\} = S \cup \{x_n\}$, where all the k carriers have exactly the same route $\pi' =< \overline{\pi} x_n >$, and let $\vec{G'}$ be the corresponding graph. The execution of $\mathcal{A}$ with injection site x_0 will have **A** perform the walk $\xi'(x_0)$ which, for the first m arcs, is identical to $\xi(x_0)$. Since **A** does not know the number of sites, it will be unable to distinguish the change: it can neither detect that one node is still missing (n is unknown) nor guarantee an exhaustive traversal (p is unknown). It will therefore stop after m moves also in $\vec{G'}$. This means that **A** stops before visiting x_n; that is, $\xi'(x_0)$ is finite but not a concrete cover, contradicting the correctness of $\mathcal{A}$. □

In other words, when the sites have unique IDs, either n or an upper-bound on the system period must be known for the problem to be solvable.

3.2. *Lower bounds on number of moves*

3.2.1. *Arbitrary routes*

We will first consider the general case, where no assumptions are made on the structure of the system routes, and establish lower bounds on the number of moves both in homogeneous and heterogeneous systems.

Theorem 3.3. *For any n, k, p, with $n \geq 9$, $\frac{n}{3} \geq k \geq 3$, and $p \geq \max\{k-1, \lceil \frac{n}{k-1} \rceil\}$, there exists a feasible* homogeneous *C-graph $\vec{G}_R$ with n sites, k carriers and period p such that $\mathcal{M}(\vec{G}_R) \geq (k-2)(p+1) + \lfloor \frac{n}{k-1} \rfloor$. This result holds even if **A** knows $\vec{G}_R$, k and p, and has unlimited memory.*

Proof. Let $S = \{s_0, \ldots, s_{n-1}\}$ and $C = \{c_0, \ldots, c_{k-1}\}$. Partition the set S into $k-1$ subsets $S_0, \ldots, S_{k-2}$ with $|S_i| = \lfloor \frac{n}{k-1} \rfloor$ for $0 \leq i \leq k-3$ and S_{k-2} containing the rest of the elements. From each set S_i select a site x_i; let $X = \{x_0, \ldots, x_{k-2}\}$. For each c_i, $i < k-1$, construct a route $\pi(c_i)$ of period p traversing S_i and such that x_i is visited only at time $t \equiv i \mod p$; this can always be done because $|S_i| \geq 3$, since $k \leq \frac{n}{3}$. Construct for c_{k-1} a route $\pi(c_{k-1})$ of period p traversing X such that it visits x_i at time $t \equiv i \mod p$ (it might visit it also at other times). Thus, by construction, carriers c_i and c_{k-1} have only one meeting point, x_i, and only at time $t \equiv i \mod p$, while $\pi(c_i)$ and $\pi(c_j)$ have no meeting points at all, $0 \leq i \neq j \leq k-2$. See Fig. 3 for an example. The agent **A** must hitch a ride with every c_i to visit the disjoint sets S_i, $0 \leq i \leq k-2$; however, **A** can enter route $\pi(c_i)$ only at time $t \equiv i \mod p$ and, once it enters it, **A** can leave it only after time p, that is only after the entire route $\pi(c_i)$ has been traversed. When traversing the last set S_i, **A** could stop as soon as all its $|S_i| \geq \lfloor \frac{n}{k-1} \rfloor$ elements are visited. Additionally **A** must perform at least $k-2$ moves on $\pi(c_{k-1})$ to reach each of the other routes. In other words, **A** must perform at least $(k-2)p + \lfloor \frac{n}{k-1} \rfloor + (k-2)$ moves. □

Costs can be significantly higher in heterogeneous systems as shown by the following.

Theorem 3.4. *For any n, k, p, with $n \geq 9$, $\frac{n}{3} \geq k \geq 3$, and $p \geq \max\{k-1, \lceil \frac{n}{k} \rceil\}$, there exists a feasible* heterogeneous *C-graph $\vec{G}_R$ with n sites, k carriers and period p such that $\mathcal{M}(\vec{G}_R) \geq (k-2)(p-1)p + \lfloor \frac{n-2}{k-1} \rfloor - 1$. This result holds even if **A** knows $\vec{G}_R$, k and p, and has unlimited memory.*

Proof. Let $C = \{c_0, \ldots, c_{k-1}\}$. Partition the set S into k subsets $S_0, \ldots, S_{k-1}$ with $|S_i| = \lfloor \frac{n-2}{k-1} \rfloor$ for $1 \leq i \leq k-1$ and S_0 containing the rest of the elements. From each set S_i $(1 \leq i \leq k-1)$ select a site x_i; let $X = \{x_1, \ldots, x_{k-1}\}$. For each c_i $(1 \leq i < k-1)$, generate a route $\pi(c_i)$ of length p traversing S_i and such that x_i is visited only at time $t \equiv i \mod p$; this can always be done because, since $k \leq \frac{n}{3}$, we have $|S_i| \geq 3$. Construct for c_0 a route $\pi(c_0)$ of period $p-1$ traversing $S_0 \cup X$ such that it visits $x_i \in X$ only at time $t \equiv i \mod (p-1)$; this can always be done since $|S_0| + |X| \geq 2 + k - 1 = k + 1$. In other words, in the system there is a route of period $p-1$, $\pi(c_0)$, and $k-1$ routes of period p, $\pi(c_i)$ for $0 < i < k$. Let **A** be at x_0 at time $t = 0$; it must hitch a ride with every c_i $(0 < i < k)$ to traverse the disjoint sets S_i; let t_i denote the first time when **A** hitches a ride with c_i. Since c_i has connection only with c_0, to catch a ride on c_i **A** must be with c_0 when it meets c_i at x_i at time t_i. To move then to a different carrier c_j $(i, j \neq 0)$, **A** must first return at x_i and hitch a ride on c_0. Since c_0 is at x_i only when $t \equiv i \mod (p-1)$ while c_i is there only when $t \equiv i \mod p$, and since $p-1$ and p are coprime, c_0 will meet c_i at time $t' > t_i$ if and only if $t \equiv t_i \mod (p(p-1))$. In other words, to move from $\pi(c_i)$ to another route $\pi(c_j)$ **A** must perform at least $p(p-1)$ moves. Since **A** must go on all routes, at least $(k-2)p(p-1)$ moves must be performed until **A** hitches a ride on the last carrier, say c_l; **A** can stop only once the last unvisited sites in $\pi(c_l)$ have been visited, i.e., after at least $\lfloor \frac{n-2}{k-1} \rfloor - 1$ additional moves. Therefore the number of moves **A** must perform is at least $(k-2)(p-1)p + \lfloor \frac{n-2}{k-1} \rfloor - 1$, completing the proof. □

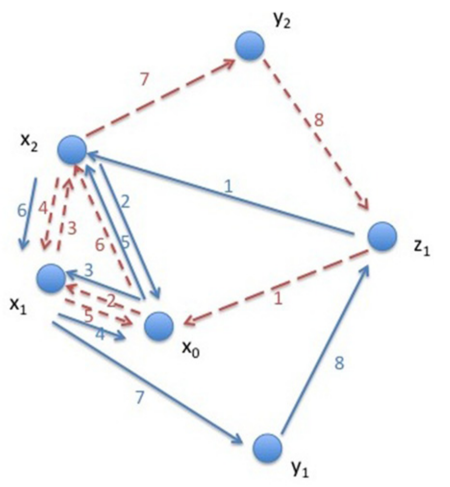

Fig. 4. CG of Theorem 3.5 with $n = 6, k = 2, \bar{m} = 3, \bar{n} = 1, p = 8$.

In other words, by Theorems 3.3 and 3.4, without any restriction on the routes, even if **A** knows n, k, p, and has unlimited memory

$$\mathcal{M}_{homo}(n, k) = \Omega(kp) \tag{1}$$

$$\mathcal{M}_{hetero}(n, k) = \Omega(kp^2). \tag{2}$$

Notice that the parameter p in the above lowerbounds can be arbitrarily large; in fact a route can be arbitrarily long even if its domain is small. This however can occur only if the carriers are allowed to go from a site x to a site y an arbitrary amount of times within the same period. Imposing restrictions on the amount of redundancy in the route the carriers must follow will clearly have an impact on the number of moves the agent needs to make.

3.2.2. Simple routes

A natural restriction is that each route is *simple*: the directed graph it describes does not contain self-loops or multi-edges; that is, $\pi(c)[i] \neq \pi(c)[i+1]$ and, if $\pi(c)[i] = \pi(c)[j]$ for $0 \leq i < j$, then $\pi(c)[i+1] \neq \pi(c)[j+1]$. If a route $\pi(c)$ is simple, then $p(c) \leq n(n-1)$. Let us stress again that even if all the routes are simple, the resulting system $\vec{G}_R$ is not necessarily simple.

The routes used in the proof of Theorems 3.3 and 3.4 were not simple. The natural question is whether simplicity of the routes can lower the cost fundamentally, i.e. to $o(kp) \subseteq o(kn^2)$ in case of homogeneous systems, and to $o(kp^2) \subseteq o(kn^4)$ in the heterogeneous ones. The answer is unfortunately negative in both cases.

We will first consider the case of homogeneous systems with simple routes.

Theorem 3.5. *For any $n \geq 4$ and $\frac{n}{2} \geq k \geq 2$ there exists a feasible* simple homogeneous *C-graph $\vec{G}_R$ with n sites and k carriers such that $\mathcal{M}(\vec{G}_R) > \frac{1}{8}kn(n-8)$. This result holds even if **A** knows $\vec{G}_R$ and k, and has unlimited memory.*

Proof. We will first construct a system satisfying the theorem's hypothesis. Let $C = \{c_1, \ldots, c_k\}$, $S = \{x_0, \ldots, x_{\bar{m}-1}, y_1, y_2, \ldots, y_k, z_1, \ldots, z_{\bar{n}}\}$, where $\bar{m} = \max\{i < n-k\colon i \text{ is prime}\}$, and let $\bar{n} = n - \bar{m} - k$. Consider the set of indices $\iota(i,j)$ defined as follows, where all operations are modulo $\bar{m}$: for $0 \leq s \leq \bar{m}-2$, $0 \leq r \leq \bar{m}-1$ and $1 \leq i \leq k$

$$\iota(i, \bar{m}s + r) = i + (s+1)r. \tag{3}$$

For simplicity, in the following we will denote $x_{\iota(i,j)}$ simply as $x(i,j)$. Finally, let the set of routes be defined as follows:

$$\pi(c_i) = \langle \mu, \delta(i), y_i \rangle \tag{4}$$

where

$$\mu = z_1, \ldots, z_{\bar{n}} \tag{5}$$

and

$$\delta(i) = x(i,1), x(i,2), \ldots, x(i, \bar{m}^2 - \bar{m}). \tag{6}$$

The system SiHo so defined is clearly homogeneous.

Claim 3.6. *In* SiHo*, for $1 \leq i \leq k$, $\pi(c_i)$ is simple and $p(c_i) = p = \bar{m}^2 - \bar{m} + 1 + \bar{n}$.*

Proof. That the value of $p(c_i)$ is as stated follows by construction. To prove simplicity we must show that each arc in the route appears only once; that is, for all $1 \le i \le k, 0 \le t' < t'' \le p-1$, if $\pi(c_i)[t'] = \pi(c_i)[t'']$ then $\pi(c_i)[t'+1] \neq \pi(c_i)[t''+1]$. This is true by construction for $t' < \bar{n}$ and $t'' \ge p-1$; i.e., for the arcs $(z_1, z_2), (z_2, z_3), \ldots, (z_{\bar{n}}, x(i,1)), (x(i, p-2), y_i), (y_i, z_1)$. Consider now the other values of t' and t''. Let $\bar{n} \le t' = \bar{m}s' + r' < \bar{m}s'' + r'' = t'' \le p-2$ with $\pi(c_i)[t'] = \pi(c_i)[t'']$; that is

$$i + (s'+1)r' \equiv i + (s''+1)r'' \mod \bar{m}. \tag{7}$$

By contradiction, let $\pi(c_i)[t'+1] = \pi(c_i)[t''+1]$; that is

$$i + (s'+1)(r'+1) \equiv i + (s''+1)(r''+1) \mod \bar{m}. \tag{8}$$

But (7) and (8) together imply that $s' \equiv s''$ (mod $\bar{m}$), which in turn (by (7)) implies that $r' \equiv r''$ (mod $\bar{m}$). However, since $\bar{m}$ is prime, this can occur only if $s' = s''$ and $r' = r''$, i.e. when $t' = t''$; a contradiction. □

Claim 3.7. *In* SiHo, *$\forall i, j$ $(1 \le i < j \le k)$, c_i and c_j meet only at the nodes of μ; this will happen whenever $t \equiv l$ (mod p), $0 \le l \le \bar{n}-1$*

Proof. By definition, the carriers meet at the nodes of μ only at the time stated by the lemma: μ is the first part of each route, and the sites in μ are different from all the others. To complete the proof we must show that two carriers will never meet anywhere else. Since y_i is only in route $\pi(c_i)$, carriers never meet there. Let us consider now the x_i's. By contradiction, let $\pi(c_i)[t] = \pi(c_l)[t]$ for some i, l, t where $1 \le i \neq l \le k, \bar{n} \le t \le p-1$; in other words, let $x(i,t) = x(l,t)$. The function ι, by definition, is such that $\iota(i+1, t) \equiv \iota(i,t) + 1 \mod \bar{m}$; since $\bar{m}$ is prime, this means that $\iota(i,j) \not\equiv \iota(l,j) \mod \bar{m}$ for $1 \le i < l \le k$ and $1 \le j \le p-1$. Therefore $\iota(i,t) \not\equiv \iota(l,t) \mod \bar{m}$; that is $x(i,t) \neq x(l,t)$: a contradiction. □

By Claims 3.6 and 3.7, the SiHo system is composed of $k \ge 2$ simple routes of period $p = \bar{m}^2 - \bar{m} + 1 - \bar{n}$, each with a distinguished site (the y_j's). The other $n-k$ sites are common to all routes; however the only meeting points in the system are those in μ and each of them is reached by all carriers simultaneously. Let **A** start at z_1 at time $t = 0$. Since only c_i can reach y_i, to visit all the distinguished sites $y_1, y_2, \ldots, y_k$, **A** must hitch a ride on all carriers. However, by Claim 3.7 carriers only connect at the points of μ, each of them reached by all carriers simultaneously. Thus, to visit y_i, **A** must hitch a ride on c_i at a site in μ at time $t \equiv f \mod p$ for some $f \in \{0, \ldots, \bar{n}-1\}$. After the visit, **A** must return to z_1, traverse all of μ hitching a ride on another carrier and follow that route until the end; only once the last distinguished site has been visited, **A** could stop, without returning to z_1. In other words, to visit each y_i (but the last), **A** will perform p moves; in the visit of the last distinguished site **A** could stop after only $p - \bar{n}$ moves; in other words, **A** needs to perform at least $(k-1)p + p - \bar{n} = kp - \bar{n}$ moves. From Claim 3.6, it follows that

$$kp - (\bar{n}) = k(\bar{m}^2 - \bar{m} + 1 + \bar{n}) - \bar{n} > k(\bar{m}^2 - \bar{m}).$$

Observe that, by definition of $\bar{m}$, we have $\frac{1}{2}(n-k-1) \le \bar{m} \le n-k-1$; furthermore, by hypothesis $k \le \frac{n}{2}$. Thus

$$\begin{aligned} k\left(\bar{m}^2 - \bar{m}\right) &\ge k\left(\frac{1}{4}(n-k-1)^2 - \frac{1}{2}(n-k-1)\right) = \frac{1}{4}k(n-k)^2 - k(n-k) + \frac{3}{4}k \\ &> \frac{1}{4}k(n-k)^2 - kn \ge \frac{1}{8}n^2k - kn \end{aligned}$$

and the theorem holds. □

Let us consider now the case of heterogeneous systems with simple routes.

Theorem 3.8. *For any $n \ge 36$ and $\frac{n}{6} - 2 \ge k \ge 4$ there exists a feasible* simple heterogeneous *C-graph $\vec{G}_R$ with n sites and k carriers such that*

$$\mathcal{M}(\vec{G}_R) \ge \frac{1}{16}(k-3)(n^2-2n)^2.$$

*This result holds even if **A** knows $\vec{G}_R$ and k, and has unlimited memory.*

Proof. To prove this theorem we will first construct a system satisfying the theorem's hypothesis. Let $C = \{c_0, \ldots, c_{k-1}\}$, $\bar{m} = \max\{q \le \frac{1}{2}(n - 3k - 4) : q \text{ is prime}\}$, and let $\bar{n} = n - 3k - 4 - 2\bar{m}$. Observe that, by definition,

$$\bar{m} \ge \left\lceil \frac{\bar{n}}{2} \right\rceil. \tag{9}$$

Partition S into six sets: $U = \{u_1, \ldots, u_{k-1}\}, V = \{v_1, \ldots, v_{k-2}\}, W = \{w_1, \ldots, w_{\bar{n}}\}, X = \{x_1, \ldots, x_{\bar{m}}\}, Y = \{y_1, \ldots, y_{\bar{m}}\}$, and $Z = \{z_1, \ldots, z_{k-1}\}$. Let the set of indices $\iota(i,j)$ be as defined in (3); for simplicity, in the following we will denote $x_{\iota(i,j)}$ and $y_{\iota(i,j)}$ simply as $x(i,j)$ and $y(i,j)$, respectively.

Let the routes $R = \{\pi(c_0), \ldots, \pi(c_{k-1})\}$ be defined as follows:

$$\pi(c_i) = < \alpha(i), \gamma(i), \delta(i), \zeta(i) > \tag{10}$$

where

$$\alpha(i) = \begin{cases} x(0,1), x(0,2), \ldots, x(0,\ \bar{m}^2 - \bar{m} - \lceil \frac{\bar{n}}{2} \rceil) & \text{for } i = 0, \\ y(i,1), y(i,2), \ldots, y(i,\ \bar{m}^2 - \bar{m} - \lfloor \frac{\bar{n}}{2} \rfloor - i + 1) & \text{for } 0 < i < k \end{cases}$$

$$\gamma(i) = \begin{cases} w_1, w_2, \ldots, w_{\lceil \frac{\bar{n}}{2} \rceil} & \text{for } i = 0 \\ w_{\lceil \frac{\bar{n}}{2} \rceil + 1}, w_{\lceil \frac{\bar{n}}{2} \rceil + 2}, \ldots, w_{\bar{n}} & \text{for } 0 < i < k \end{cases}$$

$$\delta(i) = \begin{cases} \emptyset & \text{for } i \le 1 \\ y(i,\ \bar{m}^2 - \bar{m} - \lfloor \frac{\bar{n}}{2} \rfloor - i + 2), \ldots, y(i, \bar{m}^2 - \bar{m}) & \text{for } 1 < i < k \end{cases}$$

$$\zeta(i) = \begin{cases} z_1, z_2, \ldots, z_{k-1} & \text{for } i = 0 \\ u_1, z_1, v_1, \ldots, v_{k-2} & \text{for } i = 1 \\ u_i, v_{k-2-i+2}, \ldots, v_{k-2}, z_i, v_1, \ldots, v_{k-2-i+1} & \text{for } 1 < i < k-1 \\ u_{k-1}, v_1, \ldots, v_{k-2}, z_{k-1} & \text{for } i = k-1 \end{cases}$$

and all operations on the indices are modulo $\bar{m}$. The system SiHe so defined has the following properties:

Claim 3.9. *In* SiHe, *for* $0 \le i \le k-1$, $\pi(c_i)$ *is simple, and*

$$p(c_i) = \begin{cases} \bar{m}^2 - \bar{m} + k - 1 & \text{if } i = 0 \\ \bar{m}^2 - \bar{m} + k & \text{if } 0 < i < k. \end{cases}$$

Proof. That the value of $p(c_i)$ is as stated follows by construction. To prove simplicity of $p(c_i)$ we must show that, for all $0 \le i \le k-1$ and $0 \le t' < t'' \le p(c_i) - 1$, if $\pi(c_i)[t'] = \pi(c_i)[t'']$ then $\pi(c_i)[t'+1] \ne \pi(c_i)[t''+1]$.

This is true if one or more of $\pi(c_i)[t']$, $\pi(c_i)[t'+1]$, $\pi(c_i)[t'']$, $\pi(c_i)[t''+1]$ are in $\gamma(i)$ or $\zeta(i)$. In fact, by definition, all the sites of $\gamma(i)$ and $\zeta(i)$ (Z, half the elements of W, and if $i > 0$ also $u_i \in U$) appear in $\pi(c_i)$ without any repetition, i.e., only once.

Consider now all the other cases. Let i, t', t'' ($0 \le i \le k-1$ and $0 \le t' < t'' < p(c_i) - 2$) be such that $\pi(c_i)[t'] = \pi(c_i)[t'']$ but none of $\pi(c_i)[t']$, $\pi(c_i)[t'+1]$, $\pi(c_i)[t'']$, $\pi(c_i)[t''+1]$ are in $\gamma(i)$ or in $\zeta(i)$. Let $t' = \bar{m}s' + r'$ and $t'' = \bar{m}s'' + r''$.

Let $i > 0$ (respectively, $i = 0$); that is, $\pi(c_i)[t'] = y(i, t') = y_{\iota(i,t')} = y_{i+(s'+1)r'}$ and $\pi(c_i)[t''] = y(i, t'') = y_{\iota(i,t'')} = y_{i+(s''+1)r''}$ (respectively, $\pi(c_i)[t'] = x(0, t') = x_{\iota(0,t')} = x_{(s'+1)r'}$ and $\pi(c_i)[t''] = x(0, t'') = x_{\iota(0,t''')} = x_{(s''+1)r''}$). Since $\pi(c_i)[t'] = \pi(c_i)[t'']$ it follows that $y_{i+(s'+1)r'} = y_{i+(s''+1)r''}$ (respectively, $x_{(s'+1)r'} = x_{(s''+1)r''}$); that is,

$$(s'+1)r' \equiv (s''+1)r'' \mod \bar{m}. \tag{11}$$

By contradiction, let $\pi(c_i)[t'+1] = \pi(c_i)[t''+1]$; then

$$(s'+1)(r'+1) \equiv (s''+1)(r''+1) \mod \bar{m}. \tag{12}$$

But (11) and (12) together imply that $s' \equiv s''$ (mod $\bar{m}$), which in turn implies that $r' \equiv r''$ (mod $\bar{m}$). However, since $\bar{m}$ is prime, this can occur only if $s' = s''$ and $r' = r''$, i.e. when $t' = t''$; a contradiction. □

Claim 3.10. *In* SiHe, $\forall i, j$ $(1 \le i < j \le k)$,

1. c_i *can meet with* c_0 *only at* z_i,
2. c_i and c_j *never meet.*

Proof. First observe that (1) follows by construction, since z_i is the only site in common between $\pi(c_0)$ and $\pi(c_i)$, $i > 0$. To complete the proof we must show that any other two carriers, c_i and c_j $(1 \le i < j \le k)$, will never meet; that is, $\pi(c_i)[t] \ne \pi(c_i)[t]$ for all $0 \le t \le p - 1$, where $p = p(c_i) = p(c_j) = \bar{m}(\bar{m} - 1) + k$ (by Claim 3.9).

By contradiction, let $\pi(c_i)[t] = \pi(c_j)[t] = s \in U \cup V \cup Y \cup Z \cup W$ for some $t < p$.
First observe that, by construction, c_i visits only a single distinct element of U, $u_i \ne u_j$, and only a single site in Z, $z_i \ne z_j$. Thus, $s \notin U \cup Z$.
Assume $s = v_l \in V$. By construction, $\pi(c_i)[t] = v_l$ means that $t = \bar{m}(\bar{m} - 1) + ((i + l) \mod (k-1))$; on the other hand, $\pi(c_j)[t] = v_l$ means by construction that $t = \bar{m}(\bar{m} - 1) + ((j + l) \mod (k-1))$. Thus $(i + l) \equiv (j + l) \mod (k-1)$ implying $i \equiv j \mod (k-1)$; but since $i < j \le k-1$ it follows that $i = j$, a contradiction. Hence $s \notin V$.
Assume now $s \in Y$. Let $t = \bar{m}l + r$. By definition, $\pi(c_i)[t] = \pi(c_j)[t] \in Y$ means that $y_{\iota(i,t)} = y(i, t) = \pi(c_i)[t] = \pi(c_j)[t] = y(j, t) = y_{\iota(j,t)}$. Thus $i + (l+1)r \equiv j + (l+1)r \mod \bar{m}$, that is $i \equiv j \mod \bar{m}$. This however implies $i = j$ since $i < j < k \le \bar{m}$: a contradiction. Therefore $s \notin Y$.
Finally, assume $s = w_l \in W$. By construction, $\pi(c_i)[t] = w_l$ implies that $t = \bar{m}(\bar{m} - 1) - \lfloor \frac{\bar{n}}{2} \rfloor - (i-1) + l - 2$. On the other hand, $\pi(c_j)[t] = w_l$ implies by construction that $t = \bar{m}(\bar{m} - 1) - \lfloor \frac{\bar{n}}{2} \rfloor - (j-1) + l - 2$. As a consequence, $\pi(c_i)[t] = \pi(c_i)[t] = w_l$ implies $i = j$, a contradiction. Therefore $s \notin W$.
Summarizing, $s \notin U \cup V \cup Y \cup Z \cup W$: a contradiction. □

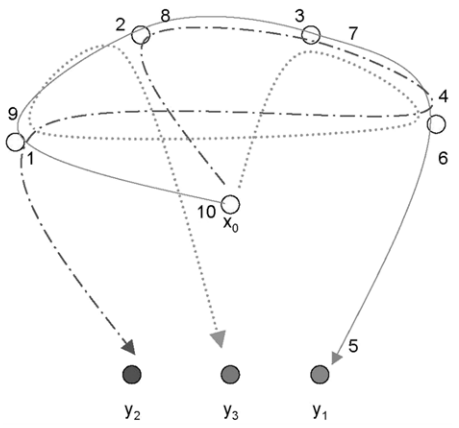

Fig. 5. CG of Theorem 3.11 with $n = 8$, $k = 3$, $p = 6$.

Given $n \geq 36$ and $\frac{n}{6} - 2 \geq k \geq 4$, let $\vec{G}_R$ be the simple graph of a SiHe system with those values. By Claims 3.9 and 3.10, in the SiHe system there is a simple route $\pi(c_0)$ of period $q = \bar{m}^2 - \bar{m} + k - 1$, and $k - 1$ simple routes $(\pi(c_i), 0 < i < k)$ of period $p = q + 1$. Each $\pi(c_i)$ with $i > 0$ has a distinguished site, u_i, not present in any other route; furthermore, $\pi(c_i)$ has no connection with $\pi(c_j)$ for $i \neq j$, while it has a unique meeting point, z_i, with $\pi(c_0)$.

Let **A** start at x_0 at time $t = 0$ with c_0. Since u_i is only in route $\pi(c_i)$, and all u_i's must be visited, **A** must hitch a ride on all c_i's.

Let t_i be the first time **A** hitches a ride on c_i at z_i. Notice that once **A** is hitching a ride on carrier c_i, since route $\pi(c_i)$ has no connection with $\pi(c_j)$, $i \neq j > 0$, to hitch a ride on c_j **A** must first return at z_i and hitch a ride on c_0. Since p and $(p - 1)$ are coprime, this can happen only at a time $t' > t_i$ such that $t' \equiv t_i \mod (qr)$; that is, after at least $p(p - 1)$ moves since **A** hitched a ride on c_i.

Since **A** must go on all routes (to visit the $u_i's$), at least $(k - 2)p(p - 1)$ moves must be performed until **A** hitches a ride on the last carrier, say c_l; then, once the last distinguished site z_l has been visited, after at least $p - (k - 1)$ moves, **A** can stop. Hence the total number of moves is at least $(k - 2)\, p\, (p - 1) + p - k + 1 > (k - 3)\, p^2$ since $p > k$.

Recall that $\bar{m}$ is the largest prime number smaller than $\frac{1}{2}(n - 3k - 4)$; since $k \leq \frac{n}{6} - 2$, we have $\bar{m} \geq \frac{1}{4}(n - 3k - 4) > \frac{n}{2}$; thus

$$p = \bar{m}^2 - \bar{m} + k > \frac{n^2}{4} - \frac{1}{2}(n - 3k - 4) + k > \frac{1}{4}(n^2 - 2n).$$

Hence the total number of moves is more than

$$(k - 3)\, p^2 > \frac{1}{16}(k - 3)(n^2 - 2n)^2 = \Omega(kn^4)$$

completing the proof. □

3.2.3. Circular routes

A further restriction on a route is to be *irredundant* (or *circular*): an arc appears in the route only once. Recall that a simple route $\pi(c)$ is *irredundant* (or *circular*) if $\vec{G}(c)$ is either a simple cycle or a simple traversal with return from a root of a tree: in other words, the undirected graph induced by the simple route is either a cycle or a tree.

By definition, any circular route $\pi(c)$ is simple, and $p(c) \leq 2(n - 1)$. The system is irredundant if all the routes are circular. Let us stress that the fact that the system is irredundant does not imply that the graph $\vec{G}_R$ is irredundant or even simple.

The graph used in the proof of Theorem 3.5 is simple but not irredundant. The natural question is whether irredundancy can lower the cost fundamentally, i.e. to $o(kp) \subseteq o(kn)$ for circular homogeneous systems and to $o(kp^2) \subseteq o(kn^2)$ for circular heterogeneous ones. The answer is unfortunately negative also in this case, as shown in the following (see Fig. 5).

Theorem 3.11. *Let the system be* homogeneous. *For any $n \geq 4$ and $\frac{n}{2} \geq k \geq 2$ there exists a feasible* irredundant *simple graph $\vec{G}_R$ with n sites and k carriers such that*

$$\mathcal{M}(\vec{G}_R) \geq n(k - 1).$$

This result holds even if **A** *knows $\vec{G}_R$, n and k, and has unlimited memory.*

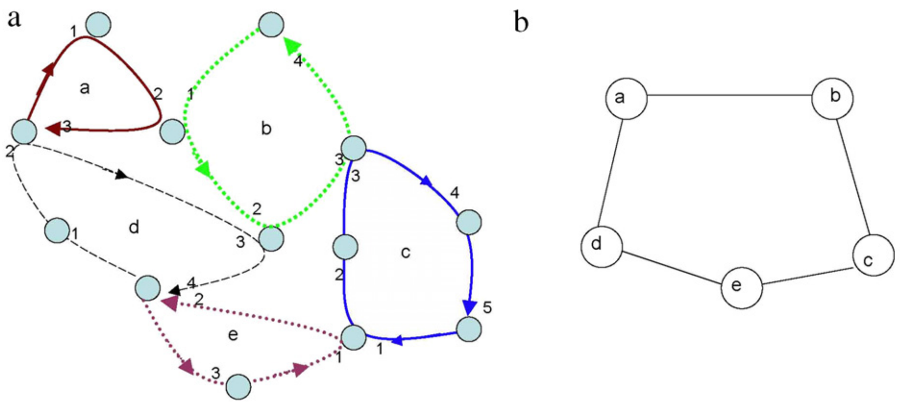

Fig. 6. (i) A circular C-Graph with carriers a, b, c, d, and e; (ii) the corresponding meeting graph. The numbers represent time.

Proof. Consider the system where $S = \{x_0, x_1, \ldots, x_{n-k-1}, y_1, y_2, \ldots, y_k\}$, $C = \{c_1, \ldots, c_k\}$, and the set of routes is defined as follows:

$$\pi(c_i) = \begin{cases} < x_0, \alpha(1), y_1, \alpha(1)^{-1} > & \text{for } i = 1 \\ < x_0, \alpha(i), \beta(i), y_i, \beta(i)^{-1}, \alpha(i)^{-1} > & \text{for } 1 < i \le k \end{cases}$$

where $\alpha(j) = x_j, x_{j+1}, x_{j+2}, \ldots, x_{n-k-1}$, $\beta(j) = x_1, x_2, \ldots, x_{j-1}$, and $\alpha(j)^{-1}$ and $\beta(j)^{-1}$ denote the reverse of $\alpha(j)$ and $\beta(j)$, respectively. In other words, the system is composed of k circular routes of period $p = 2(n-k)$, each with a distinguished site (the y_j's); the distinguished sites are reached by the corresponding carriers simultaneously at time $t \equiv n-k \mod p$. The other $n-k-1$ sites are common to all routes; however there is only a single meeting point in the system, x_0, and all carriers reach it simultaneously at time $t \equiv 0 \mod p$. More precisely, for all $1 \le i \ne j \le k$, c_i and c_j meet only at x_0; this will happen whenever $t \equiv 0 \mod p$.
Let **A** start at x_0 at time $t = 0$. To visit y_i, **A** must hitch a ride on c_i; this can happen only at x_0 at time $t \equiv 0 \mod p$; in other words, until all y_i's are visited, **A** must traverse all k routes (otherwise it will not visit all distinguished sites) returning to x_0; only once the last distinguished site, say y_j has been visited, **A** can avoid returning to x_0. Each route, except the last, takes $2(n-k)$ moves; in the last, the agent can stop after only $n-k$ moves, for a total of $2k(n-k)-(n-k)$ moves. Since $k \le \frac{n}{2}$, $2k(n-k)-(n-k) = 2nk - 2k^2 - n + k \ge (k-1)\,n$ and the theorem follows. □

We are now going to show that the cost can be order of magnitude larger if the system is not homogeneous.

Theorem 3.12. *Let the system be* heterogeneous. *For any* $0 < \epsilon < 1$, $\frac{2}{\epsilon} \le n$ *and* $2 \le k \le \epsilon\, n$, *there exists a feasible* irredundant *graph* $\vec{G}_R$ *with n sites and k carriers such that*

$$\mathcal{M}(\vec{G}_R) > \frac{1}{4}\,(1-\epsilon)^2\, n^2\,(k-2) = \Omega(n^2 k).$$

This result holds even if **A** *knows* $\vec{G}_R$, *n and k, and has unlimited memory.*

Proof. Consider a system where $S = \{x_0, \ldots, x_{q-2}, y_1, \ldots, y_{r-1}, z_1, \ldots z_{k-1}\}$, where $r < q$, and q and r are coprime, $C = \{c_0, c_1 \ldots, c_{k-1}\}$, and the set of routes is defined as follows:

$$\pi(c_i) = \begin{cases} < x_0, y_1, y_2, \ldots, y_{r-1} > & \text{for } i = 0 \\ < \alpha(i), \beta(i), z_i > & \text{for } 1 \le i < k \end{cases}$$

where $\alpha(j) = x_j, x_{j+1}, \ldots, x_{q-2}$, and $\beta(j) = x_0, \ldots, x_{j-1}$. In other words, in the system there is an irredundant route of period r, $\pi(c_0)$, and $k-1$ irredundant routes of period q, $\pi(c_i)$ for $1 \le i < k$. Each of the latter has a distinguished site (the z_i's), not present in any other route; furthermore, $\pi(c_i)$ has no connection with $\pi(c_j)$ for $i \ne j$. On the other hand, each route $\pi(c_i)$ has the same meeting point, x_0, with $\pi(c_0)$. Let t_i denote the first time c_0 and c_i meet at x_0; notice that if $i \ne j$ then $t_i \not\equiv t_j \mod (q)$. Further note that since r and q are coprime, c_0 will meet c_i at time t if and only if $t \equiv t_i \mod (q\,r)$.

Let **A** start at x_0 at time $t = 0$ with c_0. Since z_i is only in route $\pi(c_i)$, and all z_i's must be visited, **A** must hitch a ride on all c_i's. Notice that once **A** is hitching a ride on carrier c_i, since route $\pi(c_i)$ has no connection with $\pi(c_j)$, $i \ne j$, to hitch a ride on c_j **A** must first return at x_0 and hitch a ride on c_0. To hitch a ride on c_i, **A** must have been on c_0 at x_0 at some time $t' \equiv t_i \mod (qr)$; hitching again a ride on c_0 at x_0 can happen only at a time $t' < t'' \equiv t_i \mod (qr)$; in other words, after at least qr moves since **A** hitched a ride on c_i. Once on c_0 again, to hitch a ride on c_j **A** must continue to move until it reaches x_0 at time $t'' < t''' \equiv t_j \mod (qr)$, requiring at least r moves. In other words, to move from a route $\pi(c_i)$ to a different route $\pi(c_j)$ **A** must perform at least $qr + r$ moves. Since **A** must go on all routes (to visit the y_i's), at least $(k-2)(qr+r)$ moves must be performed until **A** hitches a ride on the last carrier, say c_l; then, once the last distinguished site z_l has been visited after q moves, **A** can avoid returning to a_0 and stop. Since at time $t = 0$, x is on x_0 and no other carrier is there at that time, at least $\min t_i + 1 \geq r$ moves are performed by **A** before it hitches its first ride on one of the c_i's. Hence the total number of moves is at least

$$(k-2)(qr+r)+r+q. \tag{13}$$

We now have to show how to use these facts to prove our theorem for any n and $k \leq \epsilon\, n$ $(0<\epsilon<1)$. We will consider two cases, depending on whether or not $n-k$ is even. Let $n-k$ be even; if we choose $r=\frac{n-k}{2}+1$ and $q=\frac{n-k}{2}+2$, then $n=k+q+r-3$, and r and k are coprime; hence the total number of moves is that of Expression (13). Since $k \leq \epsilon\, n$, then $n-k \geq (1-\epsilon)n$; thus

$$q\,r=\left(\frac{n-k}{2}+1\right)\left(\frac{n-k}{2}+2\right)=\left(\frac{(1-\epsilon)n}{2}+1\right)\left(\frac{(1-\epsilon)n}{2}+2\right)$$

Let $n-k$ be odd; if we choose $r=\frac{n-k+3}{2}-1$ and $q=\frac{n-k-3}{2}+1$, then $n=k+q+r-3$, and r and k are coprime. Hence the total number of moves is that of Expression (13). Since $k \leq \epsilon\, n$, then $n-k \geq (1-\epsilon)n$; it follows that

$$q\,r=\left(\frac{n-k+3}{2}-1\right)\left(\frac{n-k+3}{2}+1\right)=\left(\frac{(1-\epsilon)n+3}{2}-1\right)\left(\frac{(1-\epsilon)n+3}{2}+1\right).$$

That is, regardless of whether $n-k$ is even or odd, $q\,r > (\frac{(1-\epsilon)n}{2})^2$. Hence the total number of moves is more than

$$(k-2)\,p\,r > \frac{1}{4}(1-\epsilon)^2(k-2)\,n^2 \tag{14}$$

and the theorem holds. □

4. Optimal explorations

In this section, we show that the limitations on computability and complexity presented in the previous section are tight. In fact we prove that all necessary conditions are also sufficient and all lower bounds on costs are tight. We do so constructively presenting worst case optimal solution algorithms. An added benefit is that the algorithms are rather simple.

We will first introduce the notion of *meeting graph*, that will be useful in the description and analysis of our exploration algorithms. We will then describe and analyse two exploration algorithms, one that does not require unique node identifiers (i.e., the C-graph could be anonymous), and one for the case when distinct site IDs are available.

The *meeting graph* of a C-graph $\vec{G}$ is the undirected graph $H(\vec{G})=(C,E)$, where each node corresponds to one of the k carriers, and there is an arc between two nodes if there is at least a meeting point between the two corresponding carriers.

4.1. Exploration of anonymous C-graphs

We first consider the general problem of exploring any feasible C-graph without making any assumption on the distinguishability of the nodes. By Theorem 3.1, under these conditions the problem is not solvable if an upper bound on the periods is not known to **A** (even if **A** has unbounded memory and knows n and k).

We now prove that, if such a bound B is known, any feasible C-graph can be explored even if the graph is anonymous, the system is heterogeneous, the routes are arbitrary, and n and k are unknown to **a.** The proof is constructive: we present a simple and efficient exploration algorithm for those conditions.

Since the C-graph is anonymous and n and k are not known, to ensure that no node is left unvisited, the algorithm will have **A** explore all domains, according to a simple but effective strategy; the bound B will be used to determine termination.

Let us now describe the algorithm, Hitch-a-ride. The exploration strategy used by the algorithm is best described as a *pre-order* traversal of a *spanning-tree* of the meeting graph H, where "visiting" a node of the meeting graph H really consists of riding with the carrier corresponding to that node for B' time units, where $B'=B$ if the set of routes is known to be homogeneous, $B'=B^2$ otherwise (the reason for this amount will be apparent later).

More precisely, assume that agent **A** is riding with c for the first time; it will do so for B' time units keeping track of all new carriers encountered (list *Encounters*). By that time, **A** has not only visited the domain of c but, as we will show, **A** has encountered all carriers that can meet with c (i.e., all the neighbours of c in the meeting graph H).

At this point **A** has "visited" c in H; it will then continue the traversal of H moving to an unvisited neighbour; this is done by **A** continuing to ride with c until a new carrier c' is encountered; c will become the "parent" of c'. If all neighbours of c in H have been visited, **A** will return to its "parent" in the traversal; this is done by **A** continuing the riding with c until its parent is encountered. The algorithm terminates when **A** returns to the starting carrier and the list *Encounters* is empty.

The formal recursive description of Algorithm Hitch-a-ride is given in Fig. 7, where **A** starts with carrier c_0.

Theorem 4.1. *Algorithm* Hitch-a-ride *correctly explores any feasible C-graph $\vec{G}$ in at most $(3k-2)B'$ time where k is the number of carriers and B' is a known (upperbound on the) size of the largest route.*

Initially: $Home = c_0$; $parent(Home) := \emptyset$ $Visited := \emptyset$; $Encounters := \{c_0\}$; $N(c_0) = \emptyset$;

```
HITCH-A-RIDE($c$)

if $c = Home$ and $|Encounters| = \emptyset$ then
        Terminate
else
        if $c \notin Visited$ then
           VISIT($c$)
        end-if
        $c' \leftarrow$ GO-TO-NEXT($c$)
        HITCH-A-RIDE($c'$)
```

```
VISIT($c$)

$MyParent \leftarrow$ parent(c); $N(c) := \{MyParent\}$
ride with $c$ for $B'$ time units, and while riding
          if meet carrier $c' \notin (Encounters \cap Visited)$ then
                  $Encounters := Encounters \cup \{c'\}$
                  $N(c) := N(c) \cup \{c'\}$
          end-if
$Visited := Visited \cup \{c\}$
$Encounters := Encounters - \{c\}$
```

```
GO-TO-NEXT($c$)

if $(N(c) \cap Encounters) \neq \emptyset$ then
   Continue the ride until meet $c' \in (N(c) \cap Encounters)$
   parent-of-(c'):= c
   return $c'$
else
      Continue the ride until encountering $MyParent$
      return $MyParent$
```

Fig. 7. Algorithm HITCH-A-RIDE.

Proof. First observe that, when executing VISIT(c), **A** rides with c for B' time units, and by definition $B' \geq B \geq p(c)$; thus, **A** would visit the entire domain of c. Next observe that, after the execution of VISIT(c), $N(c)$ contains the IDs of all the carriers that have a meeting point with c. In fact, any two routes $\pi(c_i)$ and $\pi(c_j)$ that have a common meeting point will meet there every $p_{i,j}$ time units, where $p_{i,j}$ is the least common multiple of $p(c_i)$ and $p(c_j)$. If the set of routes is known to be homogeneous, by definition $\forall i, j\ B' = B \geq p_{i,j} = p(i) = p(j)$. If instead the set of routes is heterogeneous or it is homogeneous but it is not known to be so, by definition $\forall i, j\ B' = B^2 \geq p(i) \cdot p(j) \geq p_{i,j}$. Hence by riding B' time units with c, **A** will encounter all carriers that have a meeting point with c. In other words, after the "visit" of a node in H, **A** knows all its neighbours, and which ones have not yet been visited. Thus, **A** will correctly perform a pre-order visit of all the nodes of the spanning tree of H rooted in c_0 defined by the relation "parent-of". Since, as observed, the visit of a node in H consists of a visit of all the nodes in its domain, the theorem holds.

Let us now consider the cost of the algorithm. Every time routine VISIT(c) is executed, **A** performs B' moves; since a visit is performed for each carrier, there will be a total of $k \cdot B'$ moves. Routine GO-TO-NEXT(c) is used to move from a carrier c to another c' having a meeting point in common. This is achieved by riding with c until c' is met; hence its execution costs at most B' moves. The routine is executed to move from a carrier to each of its "children", as well as to return to its "parent" in the post-order traversal of the spanning tree of H defined by the relation "parent-of". In other words, it will be executed precisely $2(k-1)$ times for a total cost of at most $2(k-1)B'$ moves. The theorem then follows. $\square$

This proves that the necessary condition for *CG-Exploration* expressed by Theorem 3.1 is also sufficient. The efficiency of Algorithm HITCH-A-RIDE clearly depends on the accuracy of the upperbound B on the size p of the longest route in the system, as large values of B affect the number of moves linearly in the case of homogeneous systems, and quadratically in the case of heterogeneous system. However, it is sufficient that the upperbound is linear in p for the algorithm to be *optimal*. In fact, from Theorem 4.1 and from the lowerbounds of Theorems 3.3–3.12 we have the following theorem.

Theorem 4.2. *Let $B = O(p)$; then Algorithm* HITCH-A-RIDE *is asymptotically optimal in the worst case scenario with respect to the amount of moves. This optimality holds even if (unknowingly) restricted to the class of feasible C-graphs* with IDs, *and even if the class is further restricted to be* simple or circular *(anonymous or not).*

It is interesting to note that the amount of *memory* used by the algorithm is relatively small: $O(k \log k)$ bits are used to keep track of all the carriers and $O(\log B)$ bits to count up to B^2, for a total of $O(\log B + k \log k)$ bits.

4.2. Non-anonymous systems

We now consider the case when the nodes have distinct IDs. By Theorem 3.2, under these conditions, either n or an upperbound on the system period must be available for the exploration to be possible.

If an upperbound on the system period is available, the algorithm presented in the previous section would already solve the problem; furthermore, by Theorem 4.2, it would do so optimally. Thus, we need to consider only the situation when no upperbound on the system period is available, and just n is known.

The exploration strategy we propose is based on a *post-order* traversal of a *spanning-tree* of the meeting graph H, where "visiting" a node c of the meeting graph H now consists of riding with c for an amount of time large enough (1) to visit all the nodes in its domain, and (2) to meet every carrier that has a meeting point in common with c. In the current setting, unlike the one considered previously, an upper bound on the size of the domains is not available, making the correct termination of a visit problematic. To overcome this problem, the agent will perform a sequence of *guesses* on the largest period p, each followed by a verification (i.e., a traversal). If the verification fails, a new (larger) guess is made and a new traversal is started. The process continues until all n nodes are visited, a detectable situation since nodes have IDs.

Let us describe the strategy more precisely. Call a guess g *ample* if $g \geq P$, where $P = p$ if the graph is (known to be) homogeneous, $P = p^2$ otherwise. To explain how the process works, assume first that **A** starts the exploration riding with c_0 with an ample guess g. The algorithm would work as follows. When **A** is riding with a carrier c for the first time, it will ride (keeping track of all visited nodes) until either it encounters a new carrier c' or it has made g moves. In the first case, c becomes its "parent" and **A** starts riding with c'. In the latter, **A** has "visited" c, and will return to its parent. Termination occurs when **A** has visited n distinct nodes. With a reasoning similar to that used for the algorithm of Section 4.1, it is not difficult to see that this strategy will allow **A** to correctly explore the graph.

Observe that this strategy might work even if g is not ample, since termination occurs once **A** detects that all n nodes have been visited, and this might happen before all nodes of H have been visited. On the other hand, if the (current) guess is not ample, then the above exploration strategy might not result in a full traversal, and thus **A** might not visit all the nodes.

Not knowing whether the current guess g_i is sufficient, **A** proceeds as follows: it attempts to explore following the post-order traversal strategy indicated above, but at the first indication that the guess is not large enough, it starts a new traversal using the current carrier with a new guess $g_{i+1} > g_i$. We have three situations when the guess is discovered to be not ample: (1) while returning to its parent, **A** encounters a new carrier (the route is longer than g_i); (2) while returning to its parent, more than g_i time units elapse (the route is longer than g_i); (3) the traversal terminates at the starting carrier, but the number of visited nodes is smaller than n. In these cases the guess is doubled and a new traversal is started. Whenever a new traversal is started, all variables are reset except for the set *Visited* containing the already visited nodes.

The formal recursive description of Algorithm HITCH-A-GUESSING-RIDE is given in Fig. 8.

Theorem 4.3. *Algorithm* HITCH-A-GUESSING-RIDE *correctly explores any feasible C-graph with IDs in $O(k \cdot P)$ time provided the number of nodes is known.*

Proof. Consider the case when **A** starts the algorithm from carrier c_0 with an ample guess g. First observe that, when executing GO-TO-NEXT(c), **A** either encounters a new carrier and hitches a ride with it, or it traverses the entire domain of c (because it rides with it for $g \geq p(c)$ time units) before returning to its "parent". Moreover, while traversing c, it does encounter all the carriers it can possibly meet. In fact, any two routes $\pi(c_i)$ and $\pi(c_j)$ that have a common meeting point, will meet there every $p_{i,j}$ time units, where $p_{i,j}$ is the least common multiple of $p(c_i)$ and $p(c_j)$. If the set of routes is known to be homogeneous, by definition $\forall i, j \, g \geq p_{i,j} = p(i) = p(j)$. If instead the set of routes is heterogeneous or it is homogeneous but it is not known to be so, by definition $\forall i, j \, g \geq p(i) \cdot p(j) \geq p_{i,j}$. Hence by riding g time units with c, **A** will encounter all carriers that have a meeting point with c. In other words, when executing GO-TO-NEXT(c), if **A** does not find new carriers it "visits" a node in H, and all its neighbours but its parent have been visited. Thus, **A** will correctly perform a post-order visit of all the nodes of a spanning tree of H rooted in c_0. Hence, with an ample guess g, the visit of a node in H consists in a visit of all the nodes in its domain.

Let the current guess g_i be not ample. This fact could be detected by **A** because while returning to the parent, **A** encounters a new carrier or g_i time units elapse without encountering the parent. If this is the case, **A** will start a new traversal with the larger guess g_{i+1}. Otherwise, **A** will return to its starting carrier c and complete its "visit" of c. At this time, if all nodes have been visited, **A** will terminate (even if the guess is not ample); otherwise, a new traversal with the larger guess g_{i+1} is started. That is, if g_i is not ample and there are still unvisited nodes, **A** will start with a larger guess. Since guesses are doubled at each restart, after at most $\log P$ traversals, the guess will be ample.

Let us now consider the cost of the algorithm. First note that the worst case occurs when the algorithm terminates with an ample guess g. Let us consider such a case. Let $g_0, g_1, \ldots, g_m = g$ be the sequence of guesses leading to g and consider the number of moves performed the first time **A** uses an ample guess.

Every time routine GO-TO-NEXT(c) is executed **A** incurs in at most g_i moves. Routine GO-TO-NEXT(c) either returns a new carrier (at most k times) or a "parent" domain through routine BACKTRACK(c) (again at most k times). Routine BACKTRACK(c) spends at most g_i moves every time it is called and it is called for each backtrack (at most k times). So the overall move complexity is $3g_i \cdot k$. Let $g_0, g_1, \ldots, g_m$ be the sequence of guesses performed by the algorithm. Since the algorithm correctly terminates if a guess is ample, only g_m can be ample; that is $g_{m-1} < P \leq g_m$. Since $g_i = 2g_{i-1}$, then the total number of moves will be at most $\sum_{i=0}^{m} 3kg_i < 6kg_m = O(k \cdot P)$. □

Initially: $Home = c_0$; $parent(Home) := Visited := \emptyset$ $Encountered := \{c_0\}$.

```
HITCH-A-GUESSING-RIDE(c)

if |Visited| = n then
  Terminate
else
    c' ← GO-TO-NEXT(c)
    HITCH-A-GUESSING-RIDE(c')
```

```
GO-TO-NEXT(c) (* returns new carrier or parent *)

MyParent ← parent(c);
ride with c for g_i time units, and while riding
    let x be the current node, Visited := Visited ∪ x
    if meet carrier c' ∉ (Encountered) then
            Encountered := Encountered ∪ {c'}
            parent(c'):=c
            Return(c')
end-of-ride
if (c = Home) then
    if (|Visited| ≠ n) then
            RESTART(c)
    else
            Terminate
else
    c' ←BACKTRACK(c)
    Return(c')
```

```
BACKTRACK(c) (* backtrack unless discover guess is wrong *)

ride with c until meet MyParent
    let x be the current node, Visited := Visited ∪ x
    if while riding
          (encounter c' ∉ Encountered) or (g_i units elapse)
             RESTART(c)
end-of-ride
return MyParent
```

```
RESTART(c) (* reset variables except for Visited *)

guess := 2 · guess (** new guess**)
Home := c; parent(Home) := ∅
Encountered := {c}
HITCH-A-GUESSING-RIDE(c)
```

Fig. 8. Algorithm HITCH-A-GUESSING-RIDE.

This theorem, together with Theorem 4.1, proves that the necessary condition for *CG-Exploration* expressed by Theorem 3.2 is also sufficient.

Theorem 4.4. *Let $B = O(p)$; then Algorithm* HITCH-A-RIDE *is asymptotically optimal in the worst case scenario with respect to the amount of moves. This optimality holds even if (unknowingly) restricted to the class of* simple *feasible C-graphs* with IDs, *and even if the graphs in the class are further restricted to be* circular.

The proof follows from Theorem 4.3 and from the lowerbounds of Theorems 3.3–3.12.

Finally, notice that the amount of *memory* used by the algorithm is rather small: $O(n \log n)$ bits to keep track of all the visited nodes.

5. Concluding remarks

Developing algorithms for the exploration of highly dynamic graphs (i.e., where the network is disconnected at all times, and end-to-end travel can only exist through time and mobility) is crucial for wireless and mobile applications. The only known algorithms proposed in the literature, using random walks, have highlighted the difficulty of the task. In this paper, we have provided the first known deterministic results for the exploration of highly dynamic graphs. Although our study is limited to periodically varying graphs, this model naturally encapsulates real world networks for which no solutions existed yet. The attractiveness of our optimal solutions is further enhanced by their simplicity.

Let us stress that our model generalizes the simpler model where carrier nodes move in a free space, transfer of the agent can occur when carriers meet (i.e., they are within communication range), and the exploration is complete when all carriers have been visited. Clearly our solutions solve the problem also in this simpler model.

An interesting open problem is whether simple, yet effective, solutions also exist for other important problems (e.g., dissemination [11] and routing [12]) and general models (non-periodic).

Since the preliminary announcement of the results of this paper [21], the exploration problem in carrier networks has been further investigated. In particular, exploration with *waiting*, i.e., when the agent has the additional capability of waiting at the sites, has been studied in [20,23]; the presence of dangerous sites (e.g., black holes) [20] has been considered. Moreover, our model has been extended by adding stochastic processes [7].

Acknowledgements

We would like to thank David Ilcinkas for the helpful comments, and the anonymous referees for their useful remarks that helped improving the quality of the paper.

References

[1] S. Albers, M.R. Henzinger, Exploring unknown environments, SIAM Journal on Computing 29 (4) (2000) 1164–1188.
[2] C. Avin, M. Koucký, Z. Lotker, How to explore a fast-changing world (cover time of a simple random walk on evolving graphs), in: Proceedings of the 35th International Colloquium on Automata, Languages and Programming, ICALP, 2008, pp. 121–132.
[3] B. Awerbuch, M. Betke, M. Singh, Piecemeal graph learning by a mobile robot, Information and Computation 152 (2) (1999) 155–172.
[4] A. Balasubramanian, Y. Zhou, B. Croft, B.N. Levine, A. Venkataramani, Web search from a bus, in: Proceedings of the 2nd ACM Workshop on Challenged Networks, CHANTS, 2007, pp. 59–66.
[5] H. Baumann, P. Crescenzi, P. Fraigniaud, Parsimonious flooding in dynamic graphs, in: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing, PODC, 2009, pp. 260–269.
[6] M. A. Bender, A. Fernández, D. Ron, A. Sahai, S.P. Vadhan, The power of a pebble: exploring and mapping directed graphs, Information and Computation 176 (1) (2002) 1–21.
[7] B. Brejová, S. Dobrev, R. Královic, T. Vinar, Routing in carrier-based mobile networks, in: Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity, SIROCCO, 2011, pp. 222–233.
[8] B. Bui Xuan, A. Ferreira, A. Jarry, Computing shortest, fastest, and foremost journeys in dynamic networks, International Journal of Foundations of Computer Science 14 (2) (2003) 267–285.
[9] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: Proceedings of the 25th IEEE Conference on Computer Communications, INFOCOM, 2006, pp. 1–11.
[10] A. Casteigts, S. Chaumette, A. Ferreira, Characterizing topological assumptions of distributed algorithms in dynamic networks, in: Proceedings of the 16th Intl. Conference on Structural Information and Communication Complexity, SIROCCO, 2009, pp. 126–140.
[11] A. Casteigts, P. Flocchini, B. Mans, N. Santoro, Deterministic computations in time-varying graphs: broadcasting under unstructured mobility, in: Proceedings of the 6th IFIP International Conference on Theoretical Computer Science, TCS, 2010, pp. 111–124.
[12] A. Casteigts, P. Flocchini, B. Mans, N. Santoro, Measuring temporal lags in delay-tolerant networks, in: Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium, IPDPS, 2011, pp. 209–218.
[13] A. Casteigts, P. Flocchini, W. Quattrociocchi, N. Santoro, Time-varying graphs and dynamic networks, International Journal of Parallel, Emergent and Distributed Systems 27 (5) (2012) 387–408.
[14] A.E.F. Clementi, C. Macci, A. Monti, F. Pasquale, R. Silvestri, Flooding time of edge-Markovian evolving graphs, SIAM Journal of Discrete Mathematics, SIAMDM 24 (4) (2010) 1694–1712.
[15] R. Cohen, P. Fraigniaud, D. Ilcinkas, A. Korman, D. Peleg, Label-guided graph exploration by a finite automaton, ACM Transactions on Algorithms 4 (4) (2008).
[16] X. Deng, C.H. Papadimitriou, Exploring an unknown graph, Journal of Graph Theory 32 (3) (1999) 265–297.
[17] F. De Pellegrini, D. Miorandi, I. Carreras, I. Chlamtac, A graph-based model for disconnected ad hoc networks, in: Proceedings of the 26th IEEE Conference on Computer Communications, INFOCOM, 2007, pp. 373–381.
[18] A. Dessmark, A. Pelc, Optimal graph exploration without good maps, Theoretical Computer Science 326 (1–3) (2004) 343–362.
[19] A. Ferreira, Building a reference combinatorial model for MANETs, IEEE Network 18 (5) (2004) 24–29.
[20] P. Flocchini, M. Kellett, P. Mason, N. Santoro, Searching for black holes in subways, Theory of Computing Systems 50 (1) (2012) 158–184.
[21] P. Flocchini, B. Mans, N. Santoro, Exploration of periodically varying graphs, in: 20th International Symposium on Algorithms and Computation, ISAAC, 2009, pp. 534–543.
[22] S. Guo, S. Keshav, Fair and efficient scheduling in data ferrying networks, in: Proceedings of the 3rd ACM International Conference on Emerging Networking Experiments and Technologies, CoNext, 2007, pp. 1–12.
[23] D. Ilcinkas, A.M. Wade, On the power of waiting when exploring public transportation systems, in: Proceedings of the 15th International Conference On Principles Of Distributed Systems, OPODIS, 2011, pp. 451–464.
[24] P. Jacquet, B. Mans, G. Rodolakis, Information propagation speed in mobile and delay tolerant networks, IEEE Transactions on Information Theory 56 (10) (2010) 5001–5015.
[25] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, in: Proceedings of the 2004 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM, 2004, pp. 145–158.
[26] D. Kempe, J. Kleinberg, A. Kumar, Connectivity and inference problems for temporal networks, Journal of Computer and System Sciences 64 (4) (2002) 820–842.
[27] A. Keränen, J. Ott, DTN over areal carriers, in: Proceedings of the 4th ACM Workshop on Challenged Networks, CHANTS, 2009, pp. 67–76.
[28] V. Kostakos, Temporal graphs, Physica A 388 (6) (2009) 1007–1023.
[29] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, in: Proceedings of the 1st International Workshop on Service Assurance with Partial and Intermittent Resources, SAPIR, 2004, pp. 239–254.
[30] C. Liu, J. Wu, Scalable routing in cyclic mobile networks, IEEE Transactions on Parallel and Distributed Systems 20 (9) (2009) 1325–1338.
[31] R. O'Dell, R. Wattenhofer, Information dissemination in highly dynamic graphs, in: Proceedings of the 3rd ACM Workshop on Foundations of Mobile Computing, DIALM-POMC, 2005, pp. 104–110.
[32] CL.E. Shannon, Presentation of a maze-solving machine, in: Proc. 8th Conf. of the Josiah Macy Jr. Found. (Cybernetics), 1951, pp. 173–180.
[33] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005, pp. 252–259.
[34] J. Tang, S. Scellato, M. Musolesi, C. Mascolo, V. Latora, Small-world behavior in time-varying graphs, Physical Review E 81 (5) (2010) 055101(R).
[35] S. Wang, J.L. Torgerson, J. Schoolcraft, Y. Brenman, The deep impact network experiment operations center monitor and control system, in: Proceedings of the 3rd IEEE international Conference on Space Mission Challenges for information Technology, 2009, pp. 34–40.
[36] X. Zhang, J. Kurose, B.N. Levine, D. Towsley, H. Zhang, Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing, in: Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking, MOBICOM, 2007, pp. 195–206.
[37] Z. Zhang, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges, IEEE Communication Surveys & Tutorials 8 (1) (2006) 24–36.