Decentralised detection of periodic encounter communities in opportunistic networks

M.J. Williams[a,*], R.M. Whitaker[a], S.M. Allen[a]

[a] *Cardiff School of Computer Science & Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, UK*

Abstract

We tackle the problem of individuals being able to self-detect the encounter communities within which they periodically occur. This has widespread applicability, not least for future communication systems where content can be locally shared via wireless opportunistic networking when devices carried by participants come into close range. In this paper, we introduce a comprehensive model and decentralised algorithm to accomplish the detection of periodic communities in opportunistic networks. To the best of our knowledge, this is the first decentralised algorithm for the detection of periodic communities. We investigate the behaviour of our approach both analytically and with real-world data.

Keywords: Opportunistic network, Social network, Periodic encounter community, Decentralised detection, Temporal analysis

1. Introduction

There is now an established literature for identifying sub-structures within static network topologies, both on a centralised and decentralised basis. However, static methods fail to capture periodicity in the encounters between mobile nodes. Furthermore, in many real-world situations, periodic communities emerge from the periodic encounters between pairs of nodes. We refer to such communities as *periodic encounter communities* (PECs). The existence of PECs in a network has substantial impact on the diffusion of information among mobile nodes. Enabling nodes to detect the PECs they belong to provides them with useful context about the network they operate in.

The study of periodic encounter community detection is strongly motivated by the dominance of portable wireless devices that are carried by people and recent analyses showing the existence of multiscale human periodic encounter behaviour [1, 2, 3]. Such devices can form an *opportunistic network* [4], where data is exchanged opportunistically when devices come into close range. Devices can directly share, gain, and convey information and knowledge within PECs. Decentralised approaches are necessary in opportunistic networks due to the lack of any single repository for network information. Apart from human-based opportunistic networks, other examples of application

*Corresponding author. Tel.: +44 29 2087 4812. Fax: +44 29 2087 4598.

Email addresses: `m.j.williams@cs.cf.ac.uk` (M.J. Williams), `r.m.whitaker@cs.cf.ac.uk` (R.M. Whitaker), `stuart.m.allen@cs.cf.ac.uk` (S.M. Allen)

domains that would benefit from decentralised periodic encounter community detection include wildlife monitoring networks [5] and vehicular ad-hoc networks (VANETs) [6].

In this paper we introduce, formalise, and model the concept of periodic encounter communities, presenting to the best of our knowledge the first decentralised periodic encounter community detection algorithm. The temporal and intermittent nature of interactions between nodes makes detecting encounter communities on a decentralised basis a challenging problem. Moreover, the periodicities with which communities repeat have different time scales which may not be known a priori. The algorithm we present automatically detects the periodicities with which communities occur. Our approach combines data mining for the extraction of periodic encounter information at individual nodes with opportunistic sharing of this information when nodes are in communication range. Through opportunistic communication all nodes are able to discover the complete periodic encounter communities they appear in, including those parts of the community that a node cannot directly observe. We evaluate our approach using real-world data and explore its behaviour with a number of metrics.

The rest of the paper is organised as follows. Section 2 discusses related work in the area. In Section 3 we formulate the PEC detection problem along with its local-knowledge variant and discuss the relation between PEC detection and the periodic subgraph mining problem from the literature. Our decentralised PEC detection algorithm is presented in Section 4. In Section 5 we introduce a model for investigating the information diffusion characteristics of PECs and the limits of decentralised PEC detection. This model is applied to real-world data in Section 6 to analyse the decentralised PEC detection algorithm. Finally, we conclude in Section 7 and discuss potential future work.

2. Related work

Community detection is a well-studied problem in the field of network science. Community detection seeks to identify highly clustered components in large real-world networks. Many community detection methods have been proposed, but most are intended for offline analysis of networks (see [7] for a comprehensive survey of community detection methods). Furthermore, most methods analyse static networks; i.e., where interactions have been aggregated into a single graph regardless of their time and order. The most relevant community detection algorithms to our work are those of Hui et al. [8]. These algorithms are notable as they offer a decentralised approach for nodes to detect the static encounter communities they belong to over time. However, the algorithm considers aggregated graphs rather than any temporal or periodic trend in the encounter patterns. Other recent research into the dynamics of community structure, such as that of Palla et al. [9], has analysed the evolution of networks over time. So far there has been little work in this area that considers periodic communities.

Early analyses of human encounters focused on time-invariant characteristics, such as inter-encounter time and encounter duration [10, 11]. More recently, attention has been given to the analysis of regularity in human encounters. In particular, the work in [1] demonstrates the presence of multi-scale periodicity in a number encounter network metrics. The work of Tang et al. in [12] and [13] uses a dynamic graph representation to retain temporal information about encounters, similar to the representation used in our paper. The authors analyse the temporal dynamics of information diffusion in these graphs, but without specifically considering communities or periodicity. With a similar dynamic graph construction, Lahiri and Berger-Wolf [2] formulate the problem of identifying subgraphs that appear periodically in real-world networks. We use the framework introduced by Lahiri and Berger-Wolf to define the periodic encounter community detection problem. However, the PSE-Miner algorithm proposed by the authors is intended for use in offline analysis and assumes global knowledge of the network, and is therefore not suitable in our decentralised setting. In addition, the formulation presented by the authors does not distinguish between communities and subgraphs.

A substantial amount of related work has been motivated by the study of opportunistic networks. Such networks attempt to use encounters between wireless enabled devices to store, carry, and forward content for enabling a wide range of applications. Consequently, the temporal patterns of encounters allow content-sharing protocols in opportunistic networks to make better-informed forwarding decisions. Protocols such as those in [14, 15, 16] build an understanding of encounter familiarity between nodes. However, these protocols do not attempt to capture any regularity that may be present in encounter patterns. Some newer protocols, such as those in [17, 18], include statistical models that incorporate periodicity. These models require parameters regarding the periodicities of encounters to be known a priori. For example, in [18] a single period must be specified, which precludes detection of repeating encounters at other periods.

The aforementioned protocols analyse only pairwise patterns. Broader relationships between nodes (e.g., acquaintances of acquaintances) are not considered. Habit [19] is a protocol that attempts to merge both multi-node encounter behaviour and periodicity. Habit begins with node-centric pairwise analysis of regularity patterns between familiar strangers and, subsequently, nodes exchange their regularity patterns to build up a regularity graph. The model, however, requires a priori domain-specific period and memory parameters.

In response to these contributions, our work can make a significant contribution to applications for opportunistic networking by allowing individual devices to determine their presence within wider periodic communities. Our work also extends the concept of a community into the temporal domain.

3. The PEC detection problem

A PEC (periodic encounter community) can be thought of as a group of nodes that encounter one another periodically. The pairs of nodes in the community do not necessarily have to directly encounter one another, but may instead have an acquaintance in common with whom they are both encountering with the same periodicity. More formally, the structure of a PEC is defined in graph theoretic terms and, in particular, as a connected graph representing the nodes and their encounters. The temporal information of the PEC specifies the period with which the encounters (as represented by the graph) repeat in time and how long the pattern repeats for. We note that the same nodes may encounter with more than just one periodicity (for example, a group of nodes may meet daily during the week, and fortnightly on weekends), and thus the same set of nodes may belong to multiple PECs.

The formal definition of PECs and the language we use to discuss them are presented in detail in Section 3.1, along with a formulation of the general PEC detection problem. We present the local-knowledge variant of the PEC detection problem in Section 3.2. It is this local-knowledge PEC detection problem that we must solve in the context of opportunistic networks, since the limited connectivity and decentralised nature of these networks make it unfeasible to maintain a single source of complete knowledge of the network. Furthermore, it would be very inefficient to have nodes flood their whole (unprocessed) local encounter histories through the network to emulate a global knowledge scenario. In Section 3.3 we show that the global PECs (the result of the general

PEC detection problem) can be decomposed into multiple locally detectable PECs, and thus there is a viable solution to the local-knowledge PEC detection problem. The relationship between PEC detection and the existing problem of periodic subgraph mining is discussed in Section 3.4, along with reasons why periodic subgraph mining is not directly applicable to the local-knowledge problem.

3.1. General PEC detection formulation

PEC detection and periodic subgraph mining [2] are closely related and we adopt consistent terminology in our formulation. The representation of time in our formulation is as a series of discrete *timesteps*. The duration Q that each timestep spans is referred to as the *granularity*. In particular, for some arbitrary start time c, a timestep t spans the interval $[\ c + (t-1)Q,\ c + tQ\ )$. For simplicity, we regard encounters as discrete, zero-duration events.

Definition 1. A **simple encounter graph** $G_t = (V_t, E_t)$ is a snapshot of all encounters and nodes appearing within the time window corresponding to timestep t. That is, $\{v, u\} \in E_t$ if and only if nodes $v, u \in V_t$ were in range at least once during the time interval represented by t. A simple encounter graph $G_t = (V_t, E_t)$ is **proper** if and only if $\forall v \in V_t$ there exists $u \in V_t$ such that $\{v, u\} \in E_t$; in other words, a proper simple encounter graph is one where every node is involved in at least one encounter.

Definition 2. A **dynamic encounter graph** $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ is a time-ordered sequence of proper simple encounter graphs[1].

Definition 3. The subgraph $C = (V, E)$ of a proper simple encounter graph G_t is an **encounter community** if C is connected and $|V| > 1$.

We write $F_2 \subseteq F_1$ to denote that F_2 is a subgraph of, or equal to, F_1. We say that an encounter community C *exists* in the dynamic encounter graph $\mathcal{D}$ at timestep t if $C \subseteq G_t$. An encounter community C may exist in $\mathcal{D}$ at periodic timesteps, leading to the following definition.

Definition 4. A **periodic support set**, denoted by S_λ where $\lambda = (i, p, n)$, for an encounter community C in a dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ is a subsequence of $n > 1$ timesteps

$$S_\lambda = \langle i, i+p, i+2p, \ldots, i+(n-1)p \rangle$$

for which C exists, where $i \geq 1$ and $i + (n-1)p \leq T$. The kth timestep specified by a periodic support set, where $1 \leq k \leq n$, is given by $i + (k-1)p$ and denoted by $S_\lambda(k)$.

Given a periodic support set $S_\lambda, \lambda = (i, p, n)$ for encounter community C, we refer to n as the number of *periodic occurrences* of C specified by S_λ. We write $|S_\lambda|$ to denote periodic support set size, noting that $|S_\lambda| = n$.

Definition 5. A periodic support set $S_\lambda, \lambda = (i, p, n)$ for an encounter community C in a dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ is a **maximum periodic support set** if both $C \nsubseteq G_{i-p}$ and $C \nsubseteq G_{i+pn}$.

[1] A node exists in V_t if and only if it is involved in an encounter with another node at timestep t. The definition of dynamic graph in [2] is less strict as it permits nodes with degree zero.

Definition 6. Denoted as the pair $\langle C, S_\lambda \rangle$, a **periodic encounter community** (or **PEC**) is an encounter community C along with a maximum periodic support set S_λ for which C exists.

Note that an encounter community may exist in a dynamic encounter graph for more than one maximum periodic support set. A maximum periodic support set may be wholly contained within, intersect, or be disjoint from another. If a maximum periodic support set is contained within another, the contained periodic support set is redundant and the containing periodic support set subsumes all of the temporal information conveyed by the contained periodic support set.

Definition 7. A periodic support set S_{λ_2} is a **subset** of S_{λ_1} if and only if all timesteps contained in S_{λ_2} are contained in S_{λ_1}. Letting $\lambda_2 = (i_2, p_2, n_2)$ and $\lambda_1 = (i_1, p_1, n_1)$, an equivalent definition is that S_{λ_2} is a subset of S_{λ_1} if and only if all of the following conditions hold:

1. $i_2 \geq i_1$ and $i_2 + p_2(n_2 - 1) \leq i_1 + p_1(n_1 - 1)$
   (i.e., S_{λ_2} is temporally bounded by S_{λ_1});
2. $p_2 = kp_1$ for some integer $k > 0$
   (i.e., the period p_1 is a factor of p_2);
3. $i_2 = i_1 + lp_1$ for some integer $0 \leq l < n_1$
   (i.e., the first timestep in S_{λ_2} must be equal to a timestep in S_{λ_1}).

We denote by $S_{\lambda_2} \subseteq S_{\lambda_1}$ the relationship of S_{λ_2} being a subset of S_{λ_1}. We call S_{λ_2} a **proper subset** of S_{λ_1} if and only if $S_{\lambda_2} \subseteq S_{\lambda_1}$ and $S_{\lambda_2} \neq S_{\lambda_1}$. This relation is denoted $S_{\lambda_2} \subset S_{\lambda_1}$.

The definition of the subset relation for periodic support sets formalises the concept of temporal subsumption. If we have an encounter community C which exists in periodic support sets S_{λ_1} and S_{λ_2} such that $S_{\lambda_2} \subset S_{\lambda_1}$, then S_{λ_1} conveys more information than S_{λ_2} about the periodic occurrences of C.

Definition 8. An encounter community C' is a **subcommunity** of encounter community C if and only if $C' \subseteq C$.

Subsumption can also occur between the structural components of PECs. For example, given a PEC $\langle C, S_\lambda \rangle$, any subcommunity C' of C also exists for S_λ. If S_λ is maximum for C' then $\langle C', S_\lambda \rangle$ forms a PEC; however, in the case that $C' \subset C$ the PEC $\langle C', S_\lambda \rangle$ contains only some of the structural information conveyed by $\langle C, S_\lambda \rangle$.

Definition 9. Let $P_1 = \langle C_1, S_{\lambda_1} \rangle$ and $P_2 = \langle C_2, S_{\lambda_2} \rangle$ be two PECs. We say that P_1 is **subsumed** by P_2 if and only if $S_{\lambda_1} \subseteq S_{\lambda_2}$ and $C_1 \subseteq C_2$. We denote this relationship by $P_1 \subseteq P_2$.

Definition 10. A PEC P_1 is **maximal** if and only if there does not exist another PEC P_2, where $P_1 \neq P_2$, such that P_1 is subsumed by P_2.

Figure 1 demonstrates the subsumption and maximality criteria on an example dynamic encounter graph $\mathcal{D}$. PECs P_1 and P_2 are the only maximal PECs in $\mathcal{D}$ because they are each not subsumed by any other PEC. PECs P_3 and P_4 are examples of submaximal PECs. In particular, P_3 is structurally subsumed by P_1 due to the lack of edge $\{4, 5\}$ and P_4 is temporally subsumed by P_2 because the period of P_2 divides that of P_4 causing S_{λ_4} to be a subset of S_{λ_2}.

Maximal PECs are the fundamental PECs that we wish to extract from a dynamic encounter graph. With knowledge of all maximal PECs, all other PECs are redundant. The collection of all maximal PECs represents the most compact and complete description of the periodic encounter communities present in a dynamic encounter graph.

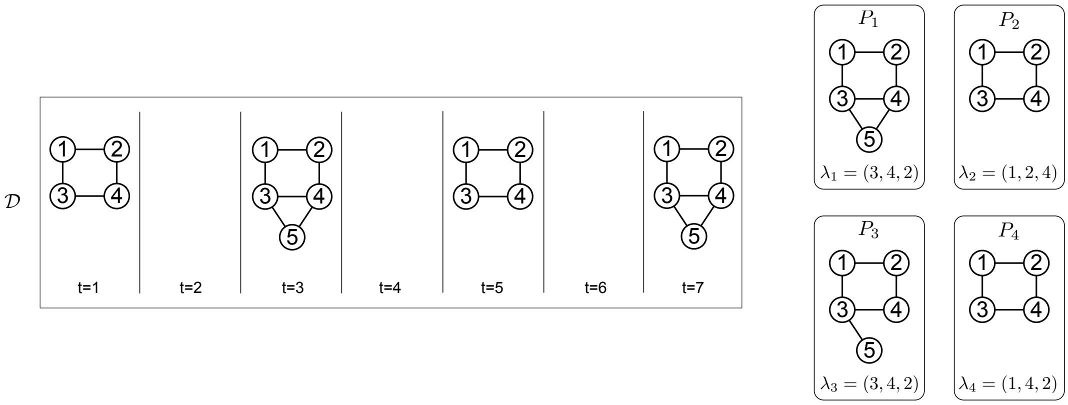

Figure 1: A dynamic encounter graph $\mathcal{D}$ and a selection of PECs in $\mathcal{D}$.

Definition 11. The **periodic encounter community detection problem** is the problem of finding all maximal periodic encounter communities that exist in a dynamic encounter graph.

3.2. Local-knowledge PEC detection formulation

The problem as introduced in Section 3.1 is presented as a *global-knowledge* problem, where mining of PECs could be carried out with the full graphs in the dynamic encounter graph available to a mining algorithm, as in [2]. Alternative to this is the node-centric perspective where the entire graph G_t is not available to any single entity. In particular, each node has only the knowledge of encounters that directly involve it. We formalise the concept of local knowledge in the following definitions.

Definition 12. For an encounter graph $G_t = (V_t, E_t)$, the **intrinsic encounter graph** $G_t^v = (V_t^v, E_t^v)$ is the subgraph of G_t induced by selecting only the edges $E_t^v = \{e \,|\, e \in E_t \wedge v \in e\}$ and their incident vertices.

Definition 13. Consider the dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$. The **intrinsic dynamic encounter graph** of a node v is the sequence of graphs $\mathcal{D}_v = \langle G_1^v, \ldots, G_T^v \rangle$.

Figure 2 shows a set of intrinsic dynamic encounter graphs and the corresponding global dynamic encounter graph. A node v's intrinsic dynamic encounter graph represents the encounter information that is directly observable by v. We note that the global encounter graph at timestep t is the aggregation of all intrinsic graphs at t; in other words, if we have dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ and denote the set of all nodes by $\mathcal{V} = V_1 \cup \ldots \cup V_T$, then

$$G_t = \bigcup_{v \in \mathcal{V}} G_t^v \ .$$

Knowledge of the (global) dynamic encounter graph is effectively distributed among the nodes in the network.

We distinguish PECs that are maximal in the global dynamic encounter graph by referring to them as *globally maximal PECs*. An *intrinsic PEC* is a PEC (be it maximal or submaximal) that exists in an intrinsic dynamic encounter graph.

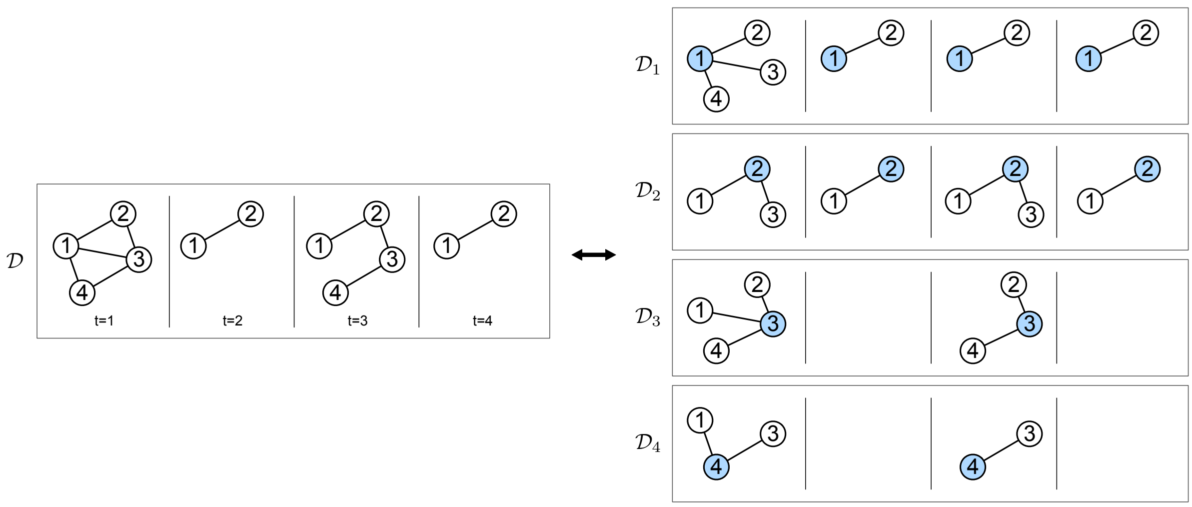

Figure 2: A global dynamic encounter graph $\mathcal{D}$ and its intrinsic dynamic encounter graphs. $\mathcal{D}_v$ denotes the intrinsic dynamic encounter graph for node v.

Definition 14. Local-knowledge periodic encounter community detection is the problem of identifying all globally maximal periodic encounter communities from local knowledge. This is a special case of the periodic encounter community detection problem (Definition 11) where no global view of the dynamic encounter graph exists. In particular, the following restrictions apply:

- **Local knowledge:** knowledge of encounters is expressed only as intrinsic dynamic encounter graphs, all of which are distributed among the corresponding nodes in the network;
- **Local exchange:** information may be exchanged between a pair of nodes only when they encounter each other.

The decentralised detection scenario corresponds to the local-knowledge problem.

3.3. Decomposition of PECs

Here we show that all globally maximal PECs decompose into intrinsic PECs. This is an important property as it means that if individual nodes extract their intrinsic PECs from their intrinsic dynamic encounter graphs, they can combine these intrinsic PECs with those of other nodes to find globally maximal PECs. Therefore, all globally maximal PECs can be detected in the local-knowledge problem.

Definition 15. A set of encounter communities $\Gamma = \{C_1, C_2, \ldots, C_m\}$ is a **community cover** of encounter community C if

$$\bigcup_{C' \in \Gamma} C' = C \ .$$

Consider the PEC $\langle C, S_\lambda \rangle$ in dynamic encounter graph $\mathcal{D}$ and a community cover Γ of C. From the definition of a PEC (Definition 6) and the subgraph property of a subcommunity, it follows that any encounter community C' in Γ exists for periodic support set S_λ. Although S_λ may not be maximum for the subcommunity C', there must exist a maximum periodic support set $S_{\lambda'}$ for C' that contains S_λ, and therefore there exists a PEC $\langle C', S_{\lambda'} \rangle$.

Definition 16. The **intrinsic cover** of encounter community $C = (V, E)$ is the set of communities $\{C_v \,|\, v \in V\}$, where $C_v = (V_v, E_v)$ is the subcommunity of C induced by selecting only the edges $E_v = \{e \,|\, e \in E \wedge v \in e\}$ and their incident vertices.

A subcommunity C' in the intrinsic cover of C corresponds to a particular node's intrinsic (i.e., local) view of C. We note that the intrinsic cover of C is also a community cover of C.

Consider a PEC $P = \langle C, S_\lambda \rangle$ and a subcommunity C' in the intrinsic cover of C. It follows that there must be an intrinsic PEC that subsumes $\langle C', S_\lambda \rangle$. Therefore, P decomposes into multiple intrinsic PECs. The same applies if P is a globally maximal PEC, and so any globally maximal PEC can be reconstructed from a local-knowledge representation.

3.4. Relation to periodic subgraph mining

The periodic subgraph mining problem introduced by Lahiri and Berger-Wolf in [2] is related to the PEC detection problem that we present in this paper. Rather than extracting periodic encounter communities as we do in our work, the periodic subgraph mining problem seeks to extract *periodic subgraph embeddings* (PSEs). A PSE in a dynamic encounter graph $\mathcal{D}$ is defined as a pair $\langle F, S_\lambda \rangle$ where F is a subgraph that exists in $\mathcal{D}$ for the periodic support set S_λ. (Subsumption and maximality rules apply to PSEs as they do to PECs.) The key distinction between a PEC and a PSE is the encounter community property of PECs. In particular, the definition of a PSE is more general as it allows subgraphs that are disconnected and subgraphs consisting of only one node.

If we assume global knowledge of the dynamic encounter graph, the PEC detection problem becomes a special case of the PSE mining problem. By extracting connected subgraphs consisting of at least two nodes from the graphs of maximal PSEs in a dynamic encounter graph, we obtain the maximal PECs. Lahiri and Berger-Wolf also show that the time and space complexity of the problem is polynomial in the size of the input dynamic encounter graph. The PSE-Miner algorithm presented as a solution to the PSE mining problem requires global knowledge, making it unsuitable for directly extracting all maximal PECs in the local-knowledge PEC detection problem. For the local-knowledge problem we instead follow a local mining and local sharing approach.

4. Decentralised PEC detection algorithm

In this section we describe our decentralised PEC detection algorithm. From the decomposition in Section 3.3 we know that global maximality of PECs can be reached from a local-knowledge representation. Therefore, the aim of the detection algorithm is to build globally maximal PECs from the local-knowledge distributed across all the nodes in the system.

4.1. Algorithm overview and parameters

Figure 3 provides an overview of the stages that a node goes through during the operation of the detection algorithm. Here we provide a brief introduction to the detection algorithm. The individual stages are described in detail later in this section. Note that the task of a node finding its local periodic communities and the periods with which these communities repeat (i.e., the task of extracting local PECs) is carried out in Stage 2. These local PECs are subsequently combined with the local PECs found by other nodes in Stage 3.

Three parameters are required during the detection algorithm. In Stage 1 the granularity Q is used. In Stage 2 p_{max} (the maximum PEC period) and n_{min} (the minimum number of periodic occurrences) are used.

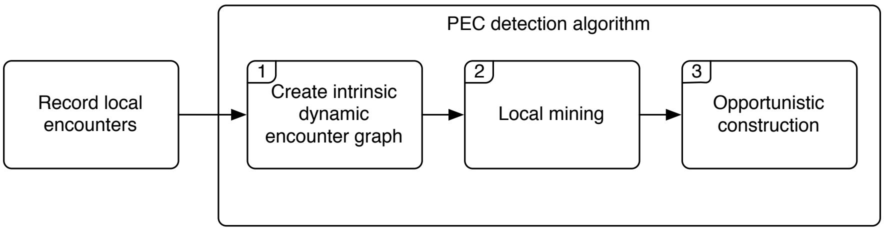

Figure 3: An overview of the stages of the PEC detection algorithm from the perspective of a node.

Before the detection algorithm is initiated, all nodes record the times of their encounters as *encounter histories* (this corresponds to the initial state in Figure 3). On initiation, the first stage of the detection algorithm is for each node to build its intrinsic dynamic encounter graph (Definition 13) from its encounter history. It is this stage where the granularity parameter (denoted by Q) is applied. As described in Section 3.1, the granularity Q is the duration of each timestep in the node's intrinsic dynamic encounter graph. The choice of granularity Q depends on the domain and application. Choosing a fine granularity results in more timesteps in the intrinsic dynamic encounter graph and so increases the computational overhead of the mining algorithm (which occurs in Stage 2). Fine granularities also have the disadvantage that the effect of small-scale randomness in the times of encounters is greater. However, in some cases we may still wish to use a fine granularity for the purpose of identifying repeating behaviour with a fine degree of temporal resolution (e.g., identifying regular encounters to within a specific hour of the day).

Details of the initiation of the PEC detection algorithm (including the building of the intrinsic dynamic encounter graph in Stage 1) and its data structures are given in Section 4.2. Note that subsequent stages of the detection algorithm only consider time in terms of timesteps in the intrinsic dynamic encounter graph. Resulting PECs are described in terms of timestep indexes rather than real-time units. (For example, the period of a PEC $\langle C, S_\lambda \rangle, \lambda = (i, p, n)$ is p timesteps.) It is trivial to convert from timesteps back to real-time units.

The node's intrinsic dynamic encounter graph built in Stage 1 is used as input to the *local mining* stage (Stage 2), which is detailed in Section 4.3. In brief, during the local mining stage individual nodes mine their intrinsic dynamic encounter graphs to obtain their intrinsic PECs. Each node implements the PSE-Miner algorithm (detailed in [2]) which extracts all (maximal intrinsic) PECs found in the node's intrinsic dynamic encounter graph. The correctness of the PSE-Miner is shown in [2] and thus we know that all PECs present in the node's intrinsic dynamic encounter graph will be identified (the criteria that define a PEC are given in Definition 6). All the attributes that constitute each PEC are automatically found by the PSE-Miner. For a PEC $P = \langle C, S_\lambda \rangle, \lambda = (i, p, n)$ these attributes are the community C, the start timestep i, the period p, and the number of periodic occurrences n. Importantly, it is the PSE-Miner that identifies the one or more periods that a community repeats with in the node's intrinsic dynamic encounter graph, resulting in one or more PECs for the community. The two parameters, $p_{max} \geq 1$ and $n_{min} \geq 2$, specified in this stage control the maximum period and minimum number of periodic occurrences, respectively. Formally, only PECs that meet the conditions $p \leq p_{max}$ and $n \geq n_{min}$ are identified. Although PECs with larger periods may exist in the intrinsic dynamic encounter graph, these are ignored.

The intrinsic dynamic encounter graph is only a local subset of the global dynamic encounter graph, and so the PECs resulting from the local mining stage (Stage 2) are not necessarily globally maximal. It is through knowledge exchange during the *opportunistic construction* stage (Stage 3) (detailed in Section 4.4) that nodes learn the globally maximal PECs they belong to. Encounters between pairs of nodes offer the opportunity for those nodes to share and expand the PECs they have discovered so far. The process of combining PECs results in PECs that have a larger community, and possibly a new period derived from the source PECs. Note that nodes will only seek to learn the PECs they are a member of.

4.2. Algorithm setup and initiation

We denote by $\mathcal{V}$ the set of all nodes in dynamic encounter graph $\mathcal{D}$. Each node $v \in \mathcal{V}$ maintains its local history of encounters with other nodes. When the detection algorithm is initiated, each node $v \in \mathcal{V}$ first builds its intrinsic dynamic encounter graph $\mathcal{D}_v = \langle G_1^v, \ldots, G_T^v \rangle$ from its encounter history. Building $\mathcal{D}_v$ is done by segmenting time into T timesteps, where each timestep represents a duration of time Q. Given some arbitrary start time c, the encounter graph G_t^v at timestep t represents v's encounters in the time interval $[\ c+(t-1)Q,\ c+tQ\ )$. The granularity Q is only used for the purpose of building the intrinsic dynamic encounter graph and is not used at any future point in the algorithm.

Each node also maintains a *knowledge base* (Definition 17), which is a data structure that holds the PECs discovered by a node so far.

Definition 17. The **knowledge base** for a node v, denoted by K_v, is a set that consists of the PECs known by v.

Knowledge bases are updated over time as locally stored PECs are combined with PECs received from other nodes. During each update, the algorithm ensures that a knowledge base K_v meets the following conditions:

1. Relevance to v: $\forall \langle C, S_\lambda \rangle \in K_v$ node v is a member of encounter community C.
2. Maximality among K_v: $\forall P_1 \neq P_2 \in K_v,\ P_1$ does not subsume P_2.

By Condition 1, a node only stores PECs that are relevant to it, and Condition 2 ensures that no redundant PECs are stored.

Once the intrinsic dynamic encounter graph $\mathcal{D}_v$ for a node v is formed and the knowledge base K_v is initialised, v then mines the intrinsic PECs from $\mathcal{D}_v$ and places them in K_v. This mining algorithm is detailed in Section 4.3. From timestep $T+1$ onwards, nodes share and update PEC information whenever they encounter each other, as detailed in Section 4.4.

The point in time to initiate mining depends on domain and application. Most applications would benefit from obtaining PEC information early; however, mining too early may result in there being too few timesteps for periodic patterns to be present.

4.3. Local mining: extraction of intrinsic PECs

In the decentralised PEC detection algorithm each node v executes the PSE-Miner algorithm [2] on its intrinsic dynamic encounter graph $\mathcal{D}_v$ to extract its (locally maximal) intrinsic PECs. As mentioned in Section 3.4, the PSE-Miner algorithm is capable of extracting all maximal PECs in a dynamic encounter graph. Therefore, if a node implements PSE-Miner, it can extract its maximal intrinsic PECs from its intrinsic dynamic encounter graph. Note that period detection is part of the mining process itself, and therefore periods do not need to be specified beforehand.

For the purpose of the PSE-Miner algorithm a dynamic encounter graph is represented as a sequence of sets of integers. To establish an invertible mapping between graphs and sets, all nodes and all edges in a dynamic encounter graph are each mapped to a unique integer label. The set representation for a particular graph $G_t = (V_t, E_t)$ is the set R_t of size $|V_t| + |E_t|$ where the integer label of each element in $V_t \cup E_t$ appears in R_t. This set representation allows fundamental operations such as graph hashing and maximal common subgraph finding to be carried out efficiently by the PSE-Miner [2].

The PSE-Miner is a single-pass algorithm. During execution the miner maintains two core data structures: a *pattern tree* and a *subgraph hash map*. As soon as a PSE ceases to be periodic it is flushed to the output stream. Those PSEs that do not have a sufficient number of periodic occurrences (n_{min}) are filtered out. A full description of the operation of the PSE-Miner algorithm, including how subgraphs and their periods are automatically identified, is provided in [2].

After a node v executes PSE-Miner on its intrinsic dynamic encounter graph, the node discards any PSEs that consist only of v (these are valid PSEs but not valid PECs). All other PSEs extracted by PSE-Miner are (locally maximal) intrinsic PECs and are therefore added to v's knowledge base.

4.4. Opportunistic construction

Opportunistic construction is the process whereby pairs of nodes share and combine their locally stored PECs when in communication range. Through repeated opportunistic construction, nodes obtain more information on the structure of the globally maximal PECs they belong to. As mentioned in Section 3.3, any non-intrinsic PEC can be obtained from its intrinsic PECs. Thus, if a construction strategy is correct and there are sufficient exchange opportunities, nodes will eventually obtain their globally maximal PECs.

When a node v encounters a node u, it receives knowledge base K_u. It is the task of v to update its own knowledge base K_v by pairwise combining the PECs in K_v with those in K_u. This update mechanism is described in Section 4.4.2. As part of knowledge base updating, node v must check if a pair of PECs are compatible to be combined to derive a new PEC. Compatibility and combination are explained in Section 4.4.1.

Note that although the local mining step returns the intrinsic PECs for a node, over time these may be subsumed by PECs generated during opportunistic construction. An intrinsic PEC subsumed by another PEC is removed since the subsuming PEC contains all the information conveyed by the intrinsic PEC. This reduces the size of knowledge bases without affecting the ability of the algorithm to build globally maximal PECs.

4.4.1. PEC compatibility and combination

Upon node v receiving a PEC P from another node, v must check which PECs in its knowledge base K_v can be combined with P to derive new PECs. A derived PEC must be connected, exist in its periodic support set, and be relevant to v.

Definition 18. Two PECs $\langle C_1, S_{\lambda_1} \rangle$ and $\langle C_2, S_{\lambda_2} \rangle$ with encounter communities $C_1 = (V_1, E_1)$ and $C_2 = (V_2, E_2)$ are **compatible** for node v if all of the following hold:

1. Relevance to v: $v \in V_1$ and $v \in V_2$;
2. Structural overlap: $E_1 \cap E_2 \neq \emptyset$;
3. Temporal containment: either $S_{\lambda_1} = S_{\lambda_2}$, $S_{\lambda_1} \subset S_{\lambda_2}$, or $S_{\lambda_2} \subset S_{\lambda_1}$.

The method of combination for two compatible PECs $\langle C_1, S_{\lambda_1}\rangle$ and $\langle C_2, S_{\lambda_2}\rangle$ depends on the direction of periodic support set containment. The case $S_{\lambda_1} = S_{\lambda_2}$ is a simple case because both C_1 and C_2 exist for the same periodic support set. In the case $S_{\lambda_1} \subset S_{\lambda_2}$, we know that C_2 exists for S_{λ_1}, but C_1 does not exist for all timesteps in S_{λ_2}. Therefore, when combining two PECs where $S_{\lambda_1} \subset S_{\lambda_2}$, the contained periodic support set (i.e., S_{λ_1}) is chosen to ensure that the resulting community exists in its support.

Formally, two compatible PECs $\langle C_1, S_{\lambda_1}\rangle$ and $\langle C_2, S_{\lambda_2}\rangle$ are combined to derive PEC $\langle C', S_{\lambda'}\rangle$ as follows:

- if $S_{\lambda_1} = S_{\lambda_2}$ then $C' = C_1 \cup C_2$ and $S_{\lambda'} = S_{\lambda_1} = S_{\lambda_2}$;
- if $S_{\lambda_1} \subset S_{\lambda_2}$ then $C' = C_1 \cup C_2$ and $S_{\lambda'} = S_{\lambda_1}$;
- if $S_{\lambda_2} \subset S_{\lambda_1}$ then $C' = C_1 \cup C_2$ and $S_{\lambda'} = S_{\lambda_2}$.

4.4.2. Knowledge base updating

During an encounter between two nodes v and u their knowledge bases K_v and K_u are exchanged[2]. For a node v receiving knowledge base K_u from node u, node v updates its own knowledge base K_v according to Algorithm 4.1.

Candidate pruning is carried out to ensure that redundant PECs are not added to the knowledge base K_v. A candidate that passes the pruning step is one that is not subsumed by any PEC already in K_v and should therefore be added to K_v. Such candidates may subsume a number of PECs already in the knowledge base. To ensure maximality among PECs in K_v, knowledge base pruning is carried out to remove any pre-existing PECs made redundant by the addition of the candidate.

5. Analysis of PEC construction using token broadcast

The time required for a global PEC to be discovered by all its constituent nodes is of primary interest for the analysis of PEC construction. It is the encounters between individual nodes that enable the information of a PEC to be shared throughout the network, and thus the patterns of these encounters have a substantial impact on the time required for a node to discover the globally maximal PECs it belongs to.

To study the spread of information in the construction of global PECs we define the equivalent scenario of *token broadcast*. Informally, token broadcast is where each node of a PEC being studied attempts to flood a unique token to all other nodes in the PEC. The route taken by a token from node u to reach node v represents the spread of u's local PEC information to v during the opportunistic construction phase of the decentralised PEC detection algorithm. The event of the token sent from u reaching v corresponds to the event of v receiving a knowledge base including some of u's PECs for the first time. The token can also represent a general packet of information, and thus the token broadcast scenario provides insight into the flow of information within a PEC.

We formally define the token broadcast scenario as follows. Consider the (global) dynamic encounter graph $\mathcal{D} = \langle G_1, G_2, \ldots, G_T\rangle$ and an arbitrary PEC $\langle C, S_\lambda\rangle$ in $\mathcal{D}$ where $\lambda = (i, p, n)$ and

[2] In practice, even before the exchange occurs a sender node can identify some PECs in its knowledge base that will not be relevant to the recipient node. Withholding these PECs reduces communication overhead. Examples include withholding a PEC that the recipient node does not appear in and withholding a PEC that has already been sent to the same node during a previous encounter.

Algorithm 4.1: KB-Update

Input: Node v whose knowledge base K_v is to be updated
Input: External knowledge base K_u
Create empty list L to hold candidate PECs
Generate candidates:
foreach $P_a \in K_v$ and $P_b \in K_u$ **do**
    if P_a and P_b are *compatible* for v and $P_b \nsubseteq P_a$ **then**
        Combine P_a and P_b to generate candidate PEC P_c
        Add P_c to L
    end
end
Prune candidates list:
foreach $P_c \in L$ and $P_a \in K_v$ **do**
    if $P_c \subseteq P_a$ **then**
        Remove P_c from L
    end
end
Prune knowledge base:
foreach $P_a \in K_v$ and $P_c \in L$ **do**
    if $P_a \subseteq P_c$ **then**
        Remove P_a from K_v
    end
end
Insert candidates:
Add all PECs in L to K_v

$C = (V, E)$. Each node v in C stores a token set T_v of received copies of tokens. We denote node v's token set after t timesteps by $T_v(t)$. Initially, each token set T_v only consists of the single token τ_v. In other words, $\forall\, v \in V,\ T_v(0) = \{\tau_v\}$. Token sharing then progresses for each timestep $i, i+1, i+2, ..., i+(n-1)p$. To carry out token sharing at timestep t, all of the encounters in the time interval for t are applied in the order they occurred. When two nodes encounter each other, each copies all of its tokens to the other. We say that full coverage has been reached in timestep t if all nodes in V have received all tokens; that is, every node v in V has $T_v(t) = \{\tau_v \mid v \in V\}$.

To characterise the broadcast of a specific PEC, only those encounters that support the PEC are used as token sharing opportunities. More specifically, only encounters corresponding to edges in E during timesteps in S_λ are used as token sharing opportunities.

5.1. Token broadcast metrics

We define the following metrics for evaluating broadcast within a PEC.

First, to quantify the extent of token spread over time we introduce metrics for token coverage. The **coverage fraction** $f_c(v, t)$ for a node v in encounter community graph $C = (V, E)$ at the end of timestep t is given by

$$f_c(v, t) = \frac{|T_v(t)| - 1}{|V| - 1} \ .$$

This measures the relative number of tokens v has obtained by the end of timestep t, excepting its own token τ_v. For a PEC $P = \langle C, S_\lambda \rangle$ where $\lambda = (i, p, n)$, we quantify the **PEC coverage** $\bar{f}_c(P, t)$ as the average coverage of nodes in C at timestep t,

$$\bar{f}_c(P, t) = \frac{1}{|V|} \sum_{v \in V} f_c(v, t) \ .$$

It is more convenient to talk in terms of the number of periodic occurrences of a PEC rather than the number of timesteps. The timestep for the kth periodic occurrence of P is given by $S_\lambda(k)$ and so we refer to $\bar{f}_c(P, S_\lambda(k))$ for the coverage fraction after k periodic occurrences.

The **broadcast time**, denoted $\Lambda(P)$, measures the number of periodic occurrences of a PEC P that were required for P to reach full coverage. $\Lambda(P)$ is equal to the smallest positive integer k such that $\bar{f}_c(P, S_\lambda(k)) = 1$. In the case that there were insufficient encounters to reach full coverage, $\Lambda(P) = \infty$.

5.2. Worst-case token broadcast time

The worst-case token broadcast time, denoted by $\Lambda_{max}(P)$, is the theoretical maximum number of periodic occurrences that an arbitrary PEC P requires to reach full coverage, under the assumption that P continues recurring indefinitely. Knowledge of the existence of a PEC $P = \langle C, S_\lambda \rangle$, where $C = (V, E)$ and $\lambda = (i, p, n)$, in dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ implies some minimum conditions on the occurrences of encounters in $\mathcal{D}$; in particular, for each edge $\{v, u\}$ in E, there must be at least one encounter between nodes v and u in each timestep $S_\lambda(1), S_\lambda(2), ..., S_\lambda(n)$. The worst-case analysis of the broadcast time for P considers the largest possible number of periodic occurrences that P would require to reach full coverage.

We note that if a PEC repeats indefinitely, the worst-case broadcast time is always finite. Since the encounters for each edge in E must occur at least once in each timestep in S_λ, if a token τ_u has not reached every node at the end of timestep $S_\lambda(k)$ then it will spread to at least one additional node in timestep $S_\lambda(k+1)$.

Definition 19. The **broadcast front**, denoted by $B_v(t)$, of token τ_v at timestep t is the set of nodes that received τ_v in timestep t but did not have it in timestep $t-1$.

A worst case for the travel of token τ_u from node u to node v is presented as follows. Consider the case where, in timestep $S_\lambda(1)$, the encounters corresponding to edges incident to u occur after all other encounters in that timestep. The effect of this is that τ_u moves one node closer to v along the shortest paths between u and v, and the broadcast front $B_u(S_\lambda(1))$ consists only of u's neighbours. If in timestep $S_\lambda(2)$ the encounters corresponding to edges incident to the nodes in $B_u(S_\lambda(1))$ occur after all other encounters, τ_u will again only move one node closer to v. If the encounters corresponding to edges incident to nodes in $B_u(S_\lambda(k))$ are always the last to occur in each timestep $S_\lambda(k+1), k = 1, \ldots, |S_\lambda|$, then the number of periodic occurrences of C required for τ_u to reach v from u is equal to the shortest path distance between v and u.

A worst-case time for a PEC to reach full coverage results when v and u are peripheral nodes, requiring a number of periodic occurrences equal to the diameter of C, denoted by $D(C)$. Thus, for a PEC $P = \langle C, S_\lambda \rangle$ we have $\Lambda_{max}(P) = D(C)$.

6. Experiments and results

In this section we evaluate decentralised PEC detection through the study of token broadcast in PECs found in a real-world encounter network. In particular, we use the encounter trace from the *MIT Reality Mining* dataset [20].

The 2004-2005 Reality Mining project carried out at the Massachusetts Institute of Technology (MIT) followed 100 subjects equipped with Bluetooth-enabled mobile phones and recorded information about their behaviour over the nine month academic period. The data collected includes Bluetooth sightings between subjects, with Bluetooth scanning carried out at five minute intervals. The long duration of the experiment permits the presence of PECs with periods in the order of hours, days, and weeks. The dataset also has the advantage of being direct encounter information between individuals, rather than inferred encounters. However, we note that Bluetooth sampling is unreliable, resulting in some missed encounters. For PEC detection, a missed encounter may result in the true PEC being temporally or structurally partitioned.

6.1. Simulating token broadcast

Simulating token broadcast on the encounter trace follows from the framework established in Section 5. When extracting the dynamic encounter graph $\mathcal{D} = \langle G_1, \ldots, G_T \rangle$ with granularity Q from the encounter trace, the orderings of actual encounters (including any repeat encounters) within each timestep $1, 2, \ldots, T$ are retained for the purpose of simulating token exchange. To simulate token broadcast for a particular globally maximal PEC $\langle C, S_\lambda \rangle$ with $C = (V, E)$ and $\lambda = (i, p, n)$, the trace is filtered so that only the encounters corresponding to edges in E and occurring during timesteps in S_λ are retained. Unique tokens are placed on the nodes and then broadcast is simulated for each timestep $i, i+1, i+2, \ldots, i+(n-1)p$. In a timestep t, each encounter from the underlying encounter trace is used as a token sharing opportunity in the order it appears during t.

	Granularity (Q)			
	4hr	**6hr**	**12hr**	**24hr**
Number of maximal PECs	509	561	897	900
Average $D(C)$	2.17	2.21	2.34	2.42
Average $\|S_\lambda\|$	4.20	4.24	4.34	4.37
Average $\Lambda(P)$	1.32	1.33	1.51	1.54
Average $\Lambda(P)/\Lambda_{max}(P)$	0.298	0.306	0.449	0.465

Table 1: Summary of PECs in the Reality Mining dataset. Four experiments were run, each with a different granularity (denoted by Q). $D(C)$ denotes community diameter, $|S_\lambda|$ denotes periodic support set size (i.e., total number of periodic occurrences of a PEC), $\Lambda(P)$ denotes broadcast time (measured in number of periodic occurrences), and $\Lambda(P)/\Lambda_{max}(P)$ gives the normalised broadcast time. PECs with $D(C) = 1$ are not included in the experiments.

6.2. Experimental setup

We set the maximum period parameter (p_{max}) to be 30 days and the minimum periodic occurrences[3] parameter (n_{min}) to be four. Other PSE-Miner parameters were left as the defaults specified in [2]; i.e., the minimum period was set to one and no timestep smoothing was carried out.

Experiments were run with granularities (denoted by Q) of 4, 6, 12, and 24 hours. Choosing a fine granularity allows the identification of periodic behaviour with greater temporal precision, but at the cost of an increase in computational overhead. Furthermore, at very fine granularities the effect of small-scale randomness in human encounter times becomes great, typically resulting in fewer PECs. Indeed, in experiments with granularity $Q = 1$ hour we found that very few PECs had periods longer than one day. The majority of PECs at this granularity were short-lived communities that repeated in consecutive timesteps for part of a day.

We note that the combination of noise in the trace dataset, the uncertain nature of human behaviour, and the crispness of our PEC definition means that PECs can become temporally fragmented. A break in encounter regularity in an encounter trace, be it due to inadequate sampling or true individual behaviour, results in a PEC either becoming temporally partitioned, structurally smaller, or not existing at all. Two or more PECs having the same encounter community, period, and phase, but spanning different durations in the trace, are assumed to be the same PEC and such duplicates were discarded from the experiments.

Finally, PECs whose communities had a diameter equal to one were not included in the analysis as these are a trivial case for PEC construction.

6.3. Results

Information on PECs obtained in the dataset is summarised in Table 1. The table shows that average diameter and average periodic support set size increase at coarser granularities. This is due to encounters being aggregated into wider snapshots, resulting in some encounter communities becoming merged. We note that in all experiments every PEC reached full coverage within the duration of time it existed.

Figure 4 shows the period and diameter of each PEC detected for granularities of 6 hours and 24 hours. We can clearly observe periodicities at one day, seven days, and 14 days, demonstrating the multiscale characteristic of human encounter behaviour. The figure also shows that many of

[3]In [2] the minimum number of periodic occurrences is denoted by σ rather than n_{min}.

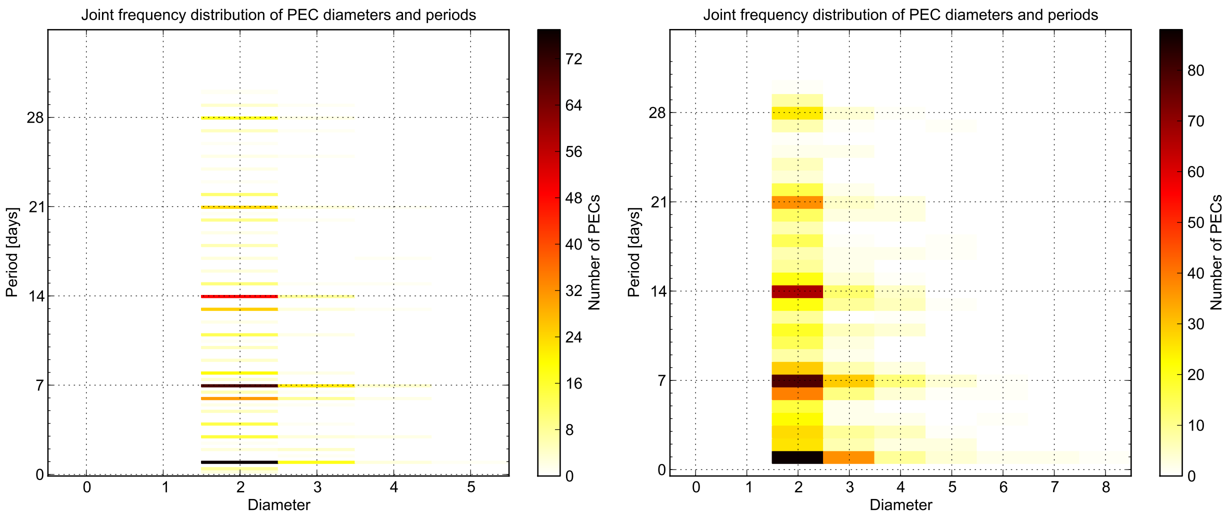

Figure 4: Joint frequency distribution of diameters and periods for PECs in the Reality Mining dataset. Left: $Q = 6$ hrs. Right: $Q = 24$ hrs.

the PECs occur at periods of one day and seven days. We suggest that this occurs because many of the PECs are students visiting the same campus on each weekday, resulting in one day PECs between Monday and Friday. Although these PECs end at the weekend, students following this weekday behaviour also exist in PECs at a period of seven days.

In Figure 5 we plotted the cumulative distribution of the normalised broadcast times of PECs. The normalised broadcast time of a PEC P is its actual broadcast time $\Lambda(P)$, normalised by its potential worst-case time $\Lambda_{max}(P)$. This quantity indicates how close a PEC's actual broadcast time is to its worst case. For granularities of 4 hours and 6 hours, 68% of PECs reached full coverage in less than 0.22 of their potential worst-case times, and 78% of PECs reached full coverage in less than 0.55 of their potential worst-case broadcast times. For the same granularities, 21% of the PECs required worst-case broadcast time.

Figure 5 also shows that coarser granularities result in PECs with broadcast times closer to their worst cases. This is reflected in the plot of community coverage over time (Figure 6). The distribution of points shows that after the first periodic occurrence, coverage was typically higher for PECs with granularity $Q = 6$ than for PECs with granularity $Q = 24$. There were a number of PECs with $Q = 24$ hours that required a 4th occurrence to reach full coverage. It appears that, although coarser granularities result in more encounters per timestep, the broadcast time still increases. We suggest that this happens because coarser granularities result in many PECs having large diameter (Figure 7). In PECs with large diameter, central nodes can have a greater negative effect on broadcast time by limiting the rate at which information spreads to the periphery of the community.

To further study the impact of diameter on broadcast time, we plotted the broadcast times for PECs grouped by diameter (Figure 8). We can see that the broadcast time increases for PECs with larger diameter. However, it is interesting that as diameter increases, PECs required worst-case broadcast time less frequently. For example, although PECs with diameter six therefore have a potential worst-case broadcast time of six, none required more than four occurrences. Only at smaller diameters do broadcast times begin to approach worst-case times; for example, 8% of PECs

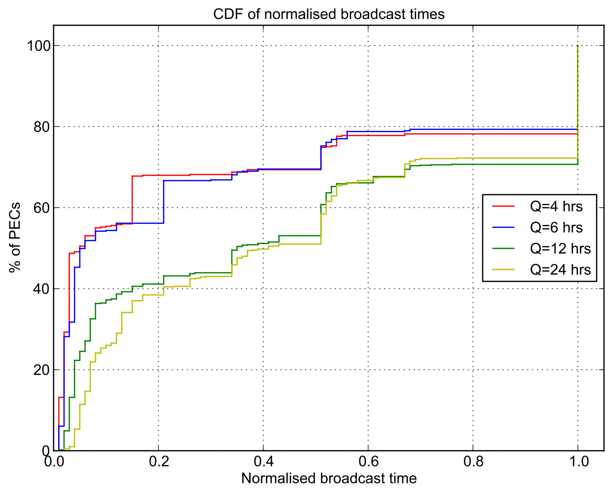

Figure 5: Cumulative distribution of normalised broadcast times for PECs in the Reality Mining dataset. The normalised broadcast time for a PEC P is given by $\Lambda(P)/\Lambda_{max}(P)$.

with diameter three required worst-case time.

The implications of these results for decentralised PEC detection are that, for most PECs in the dataset, detection of maximal PECs by the nodes in the community can occur rapidly. On average, the coverage percentage reaches 92% after the first occurrence of the community. Furthermore, the patterns of encounters within the PECs are such that, for finer granularities, the majority of the PECs were detected within 0.22 of their potential worst-case time.

7. Conclusions and future work

In this paper we defined the concept of a periodic encounter community (PEC) and the problem of nodes self-detecting PECs in an opportunistic network. To solve this problem we proposed a novel decentralised algorithm which is capable of automatically identifying community periodicities and is able to extract all globally maximal PECs, under the condition that there are sufficient exchange opportunities between nodes. Our analysis considered the diffusion of information within PECs, providing insight into the time required for PECs to be constructed.

Analytical study of diffusion in PECs shows that worst-case broadcast time for a PEC is given by its community diameter. The experimental results from a real-world dataset show that PECs with large community diameter require a longer time to reach full coverage, further demonstrating the influence of community diameter on information diffusion. Our results also show that, in the dataset we studied, the time required for a PEC reach full coverage was typically much shorter than the PEC's worst-case time.

For real-world deployment of the algorithm there are a number low-level issues that require further work. The algorithm described in this paper assumes that mining occurs once, and is then followed by opportunistic construction. In practice, nodes' encounter histories continue to grow in real time. New PECs may appear and existing PECs may cease. A simple extension would have nodes periodically erase their knowledge bases and reinitiate the algorithm. Mining can be scheduled for specific times; for example, in the context of pocket-switched networks, a convenient time is at night while the device is idle and charging. A more efficient solution would be to retain

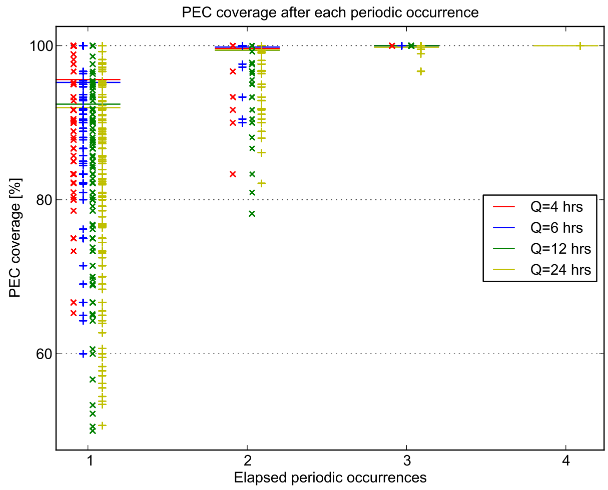

Figure 6: Coverage percentage after each periodic occurrence for PECs in the Reality Mining dataset. Points show the coverage $\bar{f}_c(P, S_\lambda(x))$ of each PEC P after each of its $x = 1, 2, \ldots, \Lambda(P)$ periodic occurrences. Horizontal lines show the average coverages.

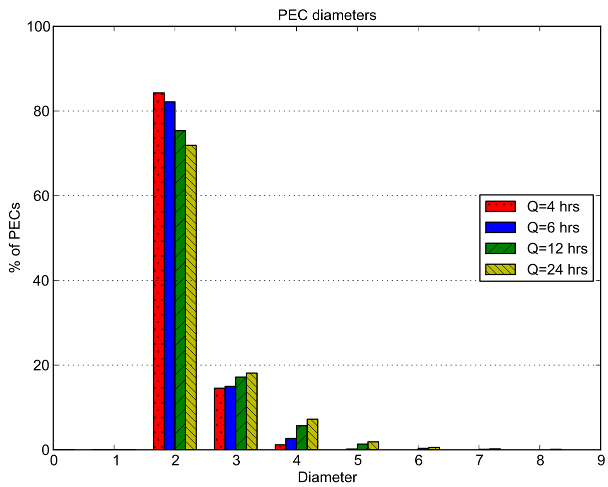

Figure 7: Diameters of PECs in the Reality Mining dataset at different granularities. PECs with diameter equal to one were not included in the experiments.

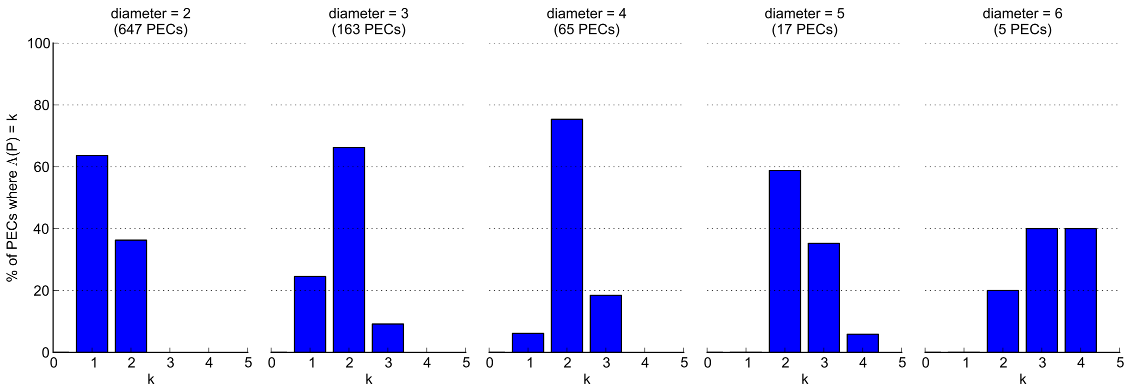

Figure 8: Comparison of PEC broadcast times (measured in number of periodic occurrences, k) by diameter. An individual plot shows the frequency distribution of broadcast times for PECs with the same diameter. $Q = 24$hrs.

the previous knowledge base when mining is reinitiated and have nodes propagate updates to PECs through the network.

Other future work will extend the model presented in this paper to a fuzzy representation. Such a representation would model uncertainty in the periodicity and structure of PECs, making them robust to noise in encounter data (i.e., missing encounters) and anomalous changes in a node's periodic patterns. Our algorithm aims to detect all PECs, including those that have stopped repeating at the time of mining. Some applications may require only the current PECs, and so a variant of the algorithm that only mines and constructs such PECs may be useful. We also note that the encounter data used in our experiments is specific to students and staff at the same academic institution. It would be interesting to study decentralised PEC detection in other scenarios, and so future work will be to consider other encounter traces from a variety of domains.

8. Acknowledgements

This work was supported by the European Commission under the SOCIALNETS (217141) FET Project. We would like to thank Nathan Eagle at MIT for providing us with the full Reality Mining dataset.

References

[1] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, On nonstationarity of human contact networks, in: Proc. 2nd Workshop on Simplifying Complex Networks for Practitioners (SIMPLEX 2010), Genoa, Italy, 2010.

[2] M. Lahiri, T. Berger-Wolf, Periodic subgraph mining in dynamic networks, Knowledge and Information Systems 24 (3) (2009) 467–497. doi:10.1007/s10115-009-0253-8.

[3] A. Clauset, N. Eagle, Persistence and periodicity in a dynamic proximity network, in: Proc. DIMACS Workshop on Computational Methods for Dynamic Interaction Networks, 2007.

[4] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, Communications Magazine, IEEE 44 (11) (2006) 134–141. doi:10.1109/MCOM.2006.248176.

[5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, SIGARCH Computer Architecture News 30 (5) (2002) 96–107. doi:10.1145/635506.605408.

[6] F. Li, Y. Wang, Routing in vehicular ad hoc networks: A survey, Vehicular Technology Magazine, IEEE 2 (2) (2007) 12–22. doi:10.1109/MVT.2007.912927.

[7] S. Fortunato, Community detection in graphs, Physics Reports 486 (3-5) (2010) 75–174. doi:10.1016/j.physrep.2009.11.002.

[8] P. Hui, E. Yoneki, S. Y. Chan, J. Crowcroft, Distributed community detection in delay tolerant networks, in: Proc. 2nd ACM/IEEE international workshop on mobility in the evolving internet architecture, ACM, Kyoto, Japan, 2007, pp. 1–8. doi:10.1145/1366919.1366929.

[9] G. Palla, A. Barabasi, T. Vicsek, Quantifying social group evolution, Nature 446 (7136) (2007) 664–667. doi:10.1038/nature05670.

[10] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on opportunistic forwarding algorithms, IEEE Transactions on Mobile Computing 6 (6) (2007) 606–620. doi:10.1109/TMC.2007.1060.

[11] P. Hui, J. Crowcroft, Human mobility models and opportunistic communications system design, Philosophical Transactions of the Royal Society A 366 (1872) (2008) 2005–2016. doi:10.1098/rsta.2008.0010.

[12] J. Tang, S. Scellato, M. Musolesi, C. Mascolo, V. Latora, Small-world behavior in time-varying graphs, Physical Review E 81 (5) (2010) 055101. doi:10.1103/PhysRevE.81.055101.

[13] J. Tang, M. Musolesi, C. Mascolo, V. Latora, Temporal distance metrics for social network analysis, in: Proc. 2nd ACM Workshop on Online Social Networks, ACM, Barcelona, Spain, 2009, pp. 31–36. doi:10.1145/1592665.1592674.

[14] C. Boldrini, M. Conti, A. Passarella, Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution, Pervasive and Mobile Computing 4 (5) (2008) 633–657. doi:10.1016/j.pmcj.2008.04.003.

[15] J. Davis, A. Fagg, B. Levine, Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks, in: Proc. 5th International Symposium on Wearable Computers, 2001, pp. 141–148. doi:10.1109/ISWC.2001.962117.

[16] A. Lindgren, A. Doria, O. Scheln, Probabilistic routing in intermittently connected networks, in: Proc. Service Assurance with Partial and Intermittent Resources (SAPIR), Fortaleza, Brazil, 2004, pp. 239–254.

[17] M. Musolesi, C. Mascolo, CAR: Context-Aware adaptive routing for Delay-Tolerant mobile networks, IEEE Transactions on Mobile Computing 8 (2) (2009) 246–260. doi:10.1109/TMC.2008.107.

[18] L. McNamara, C. Mascolo, L. Capra, Media sharing based on colocation prediction in urban transport, in: Proc. 14th ACM International Conference on Mobile Computing and Networking, ACM, San Francisco, California, USA, 2008, pp. 58–69. doi:10.1145/1409944.1409953.

[19] A. J. Mashhadi, S. B. Mokhtar, L. Capra, Habit: Leveraging human mobility and social network for efficient content dissemination in MANETs, in: Proc. of 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2009, pp. 214–219.

[20] N. Eagle, A. Pentland, D. Lazer, Inferring friendship network structure by using mobile phone data, Proceedings of the National Academy of Sciences of the United States of America 106 (36) (2009) 15274–15278. doi:10.1073/pnas.0900282106.