part of the *same* flock.

For example, in Figure 2(a) convoy query returns trajectories $\{T_1, T_2, T_3\}$ for $\theta = 3$ and for 3 time instances, while in Figure 2(b) it does not return nothing. For the moving cluster, if $\theta = 1$ then moving clusters return nothing in both Figure 2(a) and (b). On the other hand, if $\theta = 1/2$ then it returns $\{T_1, T_2, T_3\}$ in Figure 2(a) and $\{T_1, T_3, T_4\}$ in Figure 2(b), but the last one is not a convoy query. Both examples return results based on the density of the objects, but for the flock pattern it would return nothing in either examples. The reason is that in both examples the objects belong to dense areas but they do not have "strong" interaction among them.

Flock pattern query was first introduced in [5, 14], without the notion of minimum lasting time. Later [8] introduced the minimum duration as a parameter of the pattern. Unlike the convoy patterns in a flock the cluster has a predefined shape – a disk with radius r. A set of moving objects is considered a flock if there is a disk with radius r which covers all of them and there are at least some predefined number of objects in the cluster. It is shown in [8] that the discovery of the "longest" duration flock pattern is an *NP-hard* problem. As a result, [8] presents only approximation algorithms.

To the best of our knowledge our paper is the first which proposes a polynomial time solution to the flock problem with a predefined time duration. Moreover our algorithms can be applied in a streaming environment for online discovery of the flock patterns.

3. PRELIMINARIES

We assume that object O_{id} is uniquely identified by identifier id. Its movement is represented by a trajectory T_{id} which is defined as an ordered sequence of n multidimensional points $T_{id} = \{p(t_1), p(t_2), \ldots, p(t_n)\}$. Here t_i is a timestamp and $p(t_i)$ is the location of object O_{id} in the two dimensional space $\mathbb{R}^2$ as recorded at timestamp t_i ($t_i \in \mathbb{N}$, $t_{i-1} < t_i$, and $0 < i \leq n$). For simplicity when we discuss the current time instance, t_i is omitted, and we just use p_{id} to denote the object location.

Given two object locations $p_a^{t_i}$ and $p_b^{t_i}$ in a specific time instance t_i from trajectories T_a and T_b respectively, $d(p_a^{t_i}, p_b^{t_i})$ denotes the L_p distance between p_a and p_b. Even though our methods apply to any family of L_p metric distances, for ease of illustration in the rest of the paper we assume the Euclidean distance. A flock pattern query $Flock(\mu, \epsilon, \delta)$ is defined as follows:

Definition 1. *Given are a set of trajectories $\mathcal{T}$, a minimum number of trajectories $\mu > 1$ ($\mu \in \mathbb{N}$), a maximum distance $\epsilon > 0$ defined over the distance function d, and a minimum time duration $\delta > 1$ ($\delta \in \mathbb{N}$). A flock pattern $Flock(\mu, \epsilon, \delta)$ reports all sets $\mathcal{F}$ of trajectories where: for each set f_k in $\mathcal{F}$, the number of trajectories in f_k is greater than μ ($|f_k| \geq \mu$) and there exist δ consecutive time instances such that for every $t_i \in \delta$, there is a disk with center $c_k^{t_i}$ covering all $f_k^{t_i}$ points. That is: $\forall T_j \in f_k, \forall t_i \in f_k, \forall f \in \mathcal{F} : d(p_j^{t_i}, c_k^{t_i}) \leq \epsilon/2$*

The $c_k^{t_i}$ is called the center of the flock f_k at time t_i. In the above definition, a flock pattern can be viewed as a "tube" shape formed by the centers c and expanded with diameter ϵ, and having length δ (consecutive time instants) such that there are least μ trajectories which stay inside the tube all the time, as shown in Figure 3.

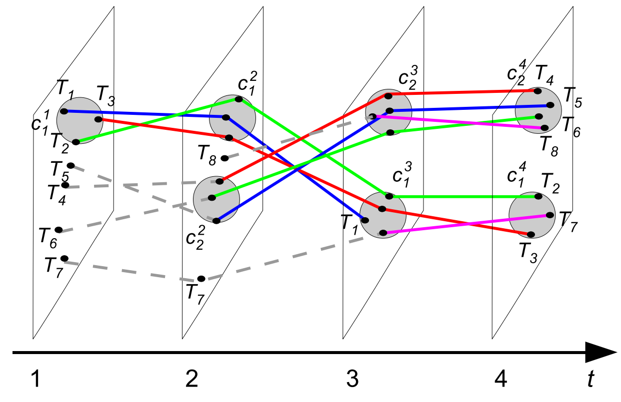

Figure 3: Flock pattern example

Having this formal definition we proceed with the complexity analysis of the flock pattern. The major challenge in this type of queries is the fact that the center of the flock pattern $c_k^{t_i}$ may not belong to any of the trajectories. Hence we cannot iterate over the discrete number of trajectory locations stored in the database and check if each one of them is a center of a flock or not. Since any point in the spatial domain can be a center of a flock there is an infinite number of possible locations to test.

Nevertheless, we show using the following Theorem that there is a limited and discrete number of locations where we can look for flocks among the infinite number of options.

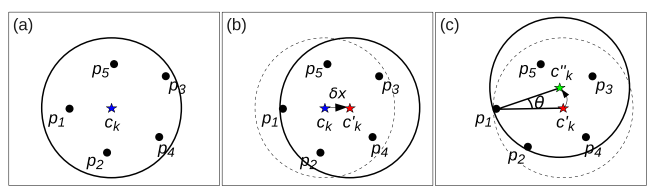

Figure 4: Finding disks to cover set of points

Theorem 1. *If for a given time instance t_i there exist a point in the space $c_k^{t_i}$ such that:*

$$\forall T_j \in f, d(p_j^{t_i}, c_k^{t_i}) \leq \epsilon/2$$

then there exists another point in the space ${c'}_k^{t_i}$ such that

$$\forall T_j \in f, d(p_j^{t_i}, {c'}_k^{t_i}) \leq \epsilon/2$$

and there are at least trajectories $T_a \in f$ and $T_b \in f$ such that

$$\forall T_j \in \{T_a, T_b\}, d(p_j^{t_i}, {c'}_k^{t_i}) = \epsilon/2$$

Theorem 1 states that if there is a disk $c_k^{t_i}$ with diameter ϵ that covers all trajectories in the flock f at time instance