R
A
P
P
O
R
T

D
E

R
E
C
H
E
R
C
H
E

L R I

COMPLEXITY FOR SELF-STABILIZING SENSOR NETWORKS

CLEMENT J / MESSIKA S / ROZOY B

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

05/2007

Rapport de Recherche N° 1472

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

Complexity for Self-Stabilizing Sensor Networks

Julien Clément, Stéphane Messika, Brigitte Rozoy
clement@lri.fr, messika@lri.fr, rozoy@lri.fr

L.R.I./C.N.R.S., Université de Paris-Sud,
bat 490, 91405 Orsay Cedex, France

abstract: Mobile sensor networks have been studied using population protocols in [2]. In this model, a collection of agents (called petrels in this paper) are described as finite state machines moving unpredictably around. They can have two kinds of interactions, only with a fixed antenna, or with each other petrels. The goal of the precedent jobs have been to compute the set of computable functions by the petrels on a set of inputs (with or without failures).In [9] we build self-stabilizing algorithms that allow to count the number of petrels. In this paper we define the random movement of the petrels and compute the complexity of these algorithms, which depends on the topology.

key words: randomized algorithms, distributed algorithms, self-stabilizing system, sensor networks, counting.
Eligible for Brief Announcement.
Eligible for Best Student Paper Award.

1 Introduction

What is the common point between zebras in Africa [13], petrels on the Great Duck island [10], children in kindergarten [1], or sequoias in Yosemite park [14]?
All of them can carry on their body a small sensor which can be useful to analyze their behavior or to preserve their integrity in hostile environments.
Research in sensor networks initiates new algorithms problematics. Indeed, classically works on distributed computing assumes that the machines in the network are powerful and have a large memory. When studying mobile sensor networks, we must consider small components, with a low computational power, and a very small amount of memory, and a short battery life. These kinds of agents can easily be modeled as very simple finite state machines.
In this paper, we study the feasibility and the efficiency of some algorithms designed for mobile sensors networks. These algorithms are based on population protocols and have been developed in [4],[3], [9] or [7]

1.1 Model

From here, we consider that the sensors are fixed on petrels. We will focus on some algorithms defined in [9]. These algorithms allow a fixed antenna to compute in a self-stabilistic way the number of petrels present in the environment.
There are several types of algorithms depending on the type of communication allowed by the sensors. In a first model they can only communicate with an antenna placed somewhere in the plan, under the condition they are close enough. In a second model they can also communicate with any other petrels they meet.
Our goal is to determine the environment on which the algorithms are efficient. In order to do so, we compute the complexity (convergence time) of the algorithms assuming that the petrels are moving like a random walk in different topologies.
So, we set the topology and the random walk followed by the petrels. On this purpose, we used classical Markov chains techniques to obtain the complexity.

1.2 Related Works

In [2], authors use population protocols to describe the evolution of the networks, they showed that in this model petrels can compute any function on the inputs on the petrels computable by a logspace Turing machine. In [7], authors tried to make these algorithms self-stabilizing, but they set some conditions on the failures, and they authorized the petrels to give wrong results if they are close to the real one. In [9], we find a way (not using population protocols), for the antenna, to compute the number of petrels. Our algorithms are self-stabilizing and give always correct results (sometimes with probability 1). All these papers deals the problem of complexity in terms of number of rounds or number of interactions, we have here more precise bounds by fixing the environment and the movement of the sensors.

1.3 Plan of the paper

In section 2 we describe the algorithms studied in this paper, and in section 3 we give principal tools to apply Markov chains theory results on our problematics, and illustrate it on a toy example.
Then, in section 4 and 5 we compute complexity bounds for several algorithms, several topologies and moves for the petrels.
We conclude in section 6.

2 Self-Stabilizing Counting Algorithms

In this section, we describe the algorithms developed in [9].

A mobile sensor network is composed of an antenna, and of n undistinguishable mobile sensors (we will use the term of petrel, instead of sensor)
A configuration of the network is given by a and $(p_1, ..., p_n)$ where a is the content of the memory of the antenna, and p_i is the state of the i^{th} petrel.
There are two kinds of events :

- the meeting of petrel number i with the antenna. After that meeting, p_i is changed, according to the protocol, to p'_i, and a to a', depending on (a, p_i) (Note that the transition is independent of i, because petrels are not distinguishable).
- the meeting of petrel number i with petrel number j. After that meeting, p_i and p_j are changed to p'_i and p'_j, depending on (p_i, p_j) (here again, independently of (i, j)).

In the Petrels-To-Antenna-Only model (TA for short), only the first kind of event is possible. i.e. the sensors do not interact with each other.
In the petrels-To-Antenna-And-To-petrels model (TATP for short), both events are possible: sensors do interact with each other.
For deterministic protocols, the last model can be divided into two sub-models, the symmetric (STATP), resp. the asymmetric one (ATATP): When two petrels meet, if their state is the same, they have to, resp. they don't have to, change to the SAME state.
Note that the asymmetric version can be viewed as probabilistic because there is a need to break the symmetry between the two petrels.
The problem can be formulate by : The number of petrels $PetrelNumbers$ is unknown from the antenna which aims at counting them.
This problem has been solved in [9] for each model in a self-stabilizing way (i.e. the value of the petrels can not be initialized). Our goal here, is to compute complexity bounds depending on the topology and on the movement on the petrels. In [9] the complexity is computed in term of rounds. A round is define by the smaller amount of interactions before every petrel meet the antenna at least once. One round could be very long depending on the movement on the petrels, here, we consider very simple probabilistic moves (random walks) and check on which topologies the algorithms are really efficients.
Let's describe some of the algorithms. See [9] for more details.

In the TA Model:
Algorithm 1 is very intuitive and simple. It converges in N rounds where N is the greatest initialization number for the petrels, but we can slightly modify it to have a convergence time of P rounds, where P is an upper bound on the number of petrels.
In the TATP Model:

Algorithm 1 Algorithm for the TA Model

```
Memory in the petrel sensors is
        number :integer
Memory at the antenna is
        registers indexed by N, initialized at 0
        PetrelNumber is max{register[i]}
When a petrel with number x approaches the antenna :
        number <- x+1
        R[number] <- R[number]+1
```

Algorithm 2 converges in P rounds.
Algorithm 3 converges in 3 rounds.

Algorithm 2 Algorithm for the ATATP model

```
Memory in the petrel sensors is
        number :integer in [1..P]
Memory at the antenna is
        T array [1..P] of boolean, initialized at 0 everywhere
        PetrelNumber is the number of i such that T[i]=1
When a petrel with number x approaches the antenna :
        T[x] <- 1
When two petrels meet :
        If their numbers are the same integer x
        then the number of one petrel becomes x+1 mod P
```

Fore more details about the algorithms and their proofs, one can read [9].

3 Introduction to Markov Chains

Modeling distributed probabilistic algorithms as Markov chains or Markov processes is a very powerful and useful tool to analyze them. It has been recently and successfully used in a lot of different topics, for example in [6], [8], [11].
The goal of this section is to give to the reader enough intuition and results to be able to understand the long-run behavior of a discrete time Markov chain and to apply it to calculate complexity bounds.
Even if we give some formals definition, the reader should have rapid access to the main results. Indeed it does not required a lot of mathematical technicalities. To help the comprehension, we will illustrate the principal results on a very simple toy example. Fore more details on the proofs one can read [5,12].

Example 1 (Toy example: Symmetric Random Walk on a segment). A random walk on a segment is a mobile agent that takes its value on $[1..N]$*.. If after* n *steps the variable is* k *then after* $n+1$ *steps the value is* $k+1$ *with probability* $\frac{1}{2}$ *and* $k-1$ *with probability* $\frac{1}{2}$*, if* $k=0$ *then the next value is always* 1 *and if* $k=N$ *the next value is* $N-1$*.*

All the notations used in the rest of section are detailed in Appendix 1

Algorithm 3 Algorithm for the STATP model

```
Memory in the petrel sensors is
        number :integer in [1..2P]
        Intention : (Keep,GiveUp)
Memory at the antenna is
        T array [1..2P] of (Free,Taken,GivenUp),
                initialized at Free everywhere
        PetrelNumber is the number of i such that T[i]=Taken
When a petrel with number x approaches the antenna :
        Depending on Intention :
        Keep :   T[x] <- Taken  /* even if T[x] was GivenUp */
        GiveUp : T[x] <- GivenUp
                 find a y such that T[y] = Free
                 T[y] <- Taken
                 number <- y
                 Intention <- Keep
When two petrels meet :
        If     their numbers are the same integer x
           and their both intentions are Keep
        Then their both intentions change to GiveUp
```

3.1 Markov Chains

Markov chains are particular classes of stochastic processes. Using sequences of independent and identically distributed random variables as stochastic models is not always efficient. Indeed, this model is not rich enough. In fact, we need more variability and Markov chains are an appropriate tool. These stochastic processes have the following fundamental property : the probabilistic dependence on the past is only related to the previous state. One can note that this property is very often verified by computer science systems, indeed it can be described as an automaton. Even if this property may seem quite simple it appears to be enough to describe a large number of complex behaviors.

Definition 2 (Markov property). let $(X_n)_{n\in\mathbb{N}}$ be a discrete time stochastic process with countable state space E. if for all integers $n \geq 0$ and all states $i_0, 1_1, \ldots, i_{n-1}, i, j$

$$\mathbb{P}\left[X_{n+1} = j / X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0\right] = \mathbb{P}\left[X_{n+1} = j / X_n = i\right] \tag{1}$$

whenever both sides of 2 are well defined, this stochastic process is called Markov chain. It is say to be an homogeneous Markov chain (HMC) if the right-handed side of 2 is independent of n.

Example 3 (Toy example). Our toy example is a Markov Chain, indeed every position is only dependent on the previous one.

The random variable X_0 is called the initial state and we note $\nu(i) = \mathbb{P}(X_0 = i)$ its probability distribution. ν is called the *initial distribution*
We are now going to describe the stability theory. We give the principal results here, but the reader may need the definitions given in Appendix 2.

Theorem 4. *Let the number of visits to state i strictly after time 0 be denoted by $N_i = \sum_{n\geq 1} \mathbb{1}_{X_n = i}$. For any state $i \in E$,*
$\mathbb{P}_i[T_i < \infty] = 1 \Leftrightarrow \mathbb{P}_i[N_i = \infty] = 1$ *and* $\mathbb{P}_i[T_i < \infty] < 1 \Leftrightarrow \mathbb{P}_i[N_i = \infty] = 0 \Leftrightarrow \mathbb{E}_i[N_i] < \infty$

In fact all these definitions are really strong properties that enable us to quote one of the main theorem of the theory of discrete homogeneous Markov chains.

Theorem 5. ***The ergodic theorem***
Let $X = (X_n)_{n\in\mathbb{N}}$ be an irreducible positive recurrent homogeneous Markov chain with the stationary distribution μ, and let $f : E \to \mathbb{R}$ be such that $\sum_{i\in E} |f(i)|\mu(i) < \infty$
Then, for any initial distribution ν, $\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} f(X_k) = \sum_{i\in E} f(i)\mu(i)$

This result allows us to describe the convergence of the long-run behavior of the homogeneous Markov chain. We will need some corollaries specific to the number of returns.

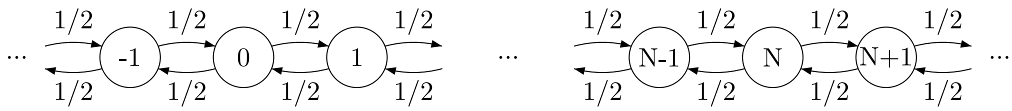
Corollary 6. *Under the notations and hypothesis of the ergodic theorem 5:* $\mathbb{E}_0[T_0] = \frac{1}{\mu(0)}$

Corollary 7. *Under the notations and hypothesis of the ergodic theorem 5:* $\mathbb{E}_0[\sum_{k=1}^{n} \mathbb{1}_{\{X_k=0\}}] \underset{n\to\infty}{\sim} n\mu(0)$

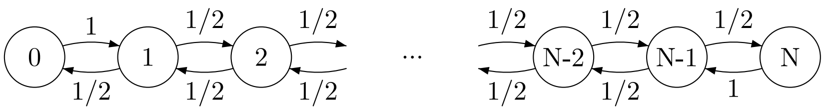
3.2 Movement of the Petrels

We choose that the petrels will evolve in dimension 1 or 2 and in bounded or unbounded environment. In each of this 4 models we can associate a random walk (Markov chain) describing the movement of one petrel. We describe here the simplest ones, all the random walks here are symmetric.

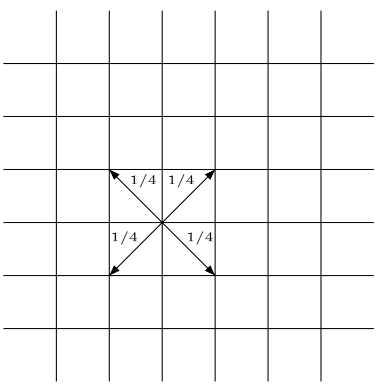
In dimension 1, Unbounded This movement follows the classical random walk on $\mathbb{Z}$



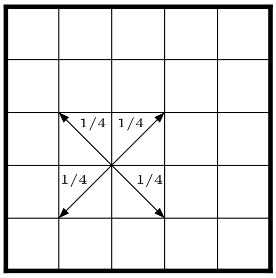
In dimension 1, Bounded This movement is a random walk with reflecting barriers, i.e. when the sensor reaches one of the bounds it goes back with probability 1.



In dimension 2, Unbounded The movement here is a random walk on $\mathbb{Z}^2$ with four possible elementary steps : NorthEast (NE), NorthWest (NW), SouthEast (SE) and SouthWest (SW) each of them have probability $\frac{1}{4}$.



In dimension 2, Bounded The movement is the same as in the previous paragraph but, again when the sensors reach one of the bounds it comes back from its initial direction with probability 1.



4 The To-Antenna (TA) Model

4.1 TA Model in Dimension 1

Unbounded Environment. In this section, the petrels move following the figure in section 3.2.
We call X_n the position of the petrel after n steps of the random walk.
We suppose the antenna fixed at the site 0, and we need to compute the number of steps needed to visit enough this site such that the algorithms converge. In order to achieve this we must find out the properties of the state 0 of the HMC (Is it recurrent ? Is it positive recurrent ?)

Proposition 8. *The symmetric random walk on $\mathbb{Z}$ is null recurrent.*

The proof this proposition is done in appendix 3 and can also be found in [5], [12].

Complexity for Algorithm 1 In dimension 1, in an unbounded environnement, the previous result shows that Algorithm 1 does not converge in a finite number of steps. The expected time before a return to the antenna is infinite. We can not use this algorithm in such an environment, and with such a movement. Thus, we have the following result.

Proposition 9. *The convergence time for the algorithm* 1 *in an* 1*-dimension unbounded environment is infinite.*

Several ideas can help to solve this problem, for example, one can imagine that the probability to get back to the antenna changes with your position.
Here, we choose to adopt a more realistic model in which the petrels do not evolve in an unbounded environment. We fix the size of the environment by placing (reflecting) barriers such that if a petrel reaches the barrier it automatically goes back.
Here are the results for this model in dimension 1.

Bounded Environment . One now studies a random walk on the interval $\{0, 1, 2, \ldots, N\}$ with reflective barriers. The movement in this section is described in section 3.2.

Proposition 10. *The Markov chain $X = (X_n)_{n\in\mathbb{N}}$ is positive recurrent.*

Proof. It is easily noticed that all the points communicate. There is thus only one class of communication and the chain is irreducible. E being finished, the Markov chain $X = (X_n)_{N\in\mathbb{N}}$ is recurrent positive.

One deduces from this proposal that there is a unique invariant probability μ. Our goal is to determine the value of $\mu(0)$. Then using the corollary 6 we will be able to find $\mathbb{E}[T_0]$. In order to determine it, one writes: $\forall x \in E \quad \mu(x) = \sum_{y\in E} \mu(y)M(y,x)$ Therefore, we get to the following system :

$$\begin{cases} \mu(x) = \frac{1}{2}\mu(x-1) + \frac{1}{2}\mu(x+1) \text{ if } 2 \le x \le n-2 \\ \mu(1) = \mu(0) + \frac{1}{2}\mu(2) \\ \mu(0) = \frac{1}{2}\mu(1) \\ \mu(N) = \frac{1}{2}\mu(N-1) \\ \mu(N-1) = \frac{1}{2}\mu(N-2) + \mu(N) \end{cases}$$

So, writing $\mu(x) = \frac{1}{2}\mu(x) + \frac{1}{2}\mu(x)$ in the first equality, we have, for $2 \le z \le N-2$, $\mu(z+1) - \mu(z) = \frac{p}{q}[\mu(z) - \mu(z-1)]$

After summing we get, for $2 \le x \le N-2$, $\mu(x+1) - \mu(2) = \sum_{z=2}^{x}[\mu(z) - \mu(z-1)] = [\mu(x) - \mu(1)]$ i.e:

$\mu(x+1) = 2(\frac{1}{2} - 1)\mu(0) + \mu(x) - 2\mu(0)$

So, $\mu(x+1) = \mu(x)$ if $2 \le x \le N-2$

We recall that μ is a probability, therefore it shall be of mass 1. It then remains us to determine the mass of this measure.

$\mu(0) = \frac{1}{2}\mu(1)$ and $\mu(N) = \frac{1}{2}\mu(N-1) = \frac{1}{2}\mu(1)$

Then, $\mu(E) = (N-1)\mu(1)$. We can easily deduce $\mu(0)$: $\mu(0) = \frac{1}{2(N-1)}$

Again, we use corollary 6 in order to find the time to return to 0, we obtain $\mathbb{E}_0[T_0] = 2(N-1)$

And corollary 7 gives us: $\mathbb{E}_0[\sum_{k=1}^{n} \mathbb{1}_{\{X_k=0\}}] \underset{n\to\infty}{\sim} \dfrac{n}{2(N-1)}$

Complexity for Algorithm 1 We are now able to study the complexity of the algorithm 1. Indeed, we prove in [9] that algorithm 1 converges in P rounds, where P is the number of petrels.

Proposition 11. *Under the hypothesis that sensors move following a random walk on a straight line with reflective barriers, algorithm* 1 *converges in $O(2PN)$ steps where N is the size of the segment and P, a bound on the number of petrels.*

Proof: The number of returns to the antenna in n steps is equivalent to $\frac{n}{2(N-1)}$, so if $n = 2PN$ we have more than P returns which ensures the convergence of algorithm 1.

4.2 The TA Model in Dimension 2

Unbounded Environment .

The movement of the petrels in dimension 2 is described in section 3.2. As usual, we define X_n the random variable equal to the position of the petrel at the moment n. It is supposed that $X_0 = (0,0)$, and we suppose the antenna at the position $(0,0)$. We also define $O_j = \mathbb{1}_{\{Y_j=0\}}$ The reasoning is identical to the one carried out in dimension 1. One thus obtains

$$\begin{cases} \mathbb{P}_0(O_{2j+1} = 1) = 0 \\ \mathbb{P}_0(O_{2j} = 1) = \frac{C_{2j}^{j}}{4^j} \times \frac{C_{2j}^{j}}{4^j} \end{cases}$$

We can deduce that $\mathbb{E}_0[N_0] = \sum_{n\ge 1}\left(\dfrac{C_{2j}^{j}}{4^j}\right)^2$, then $\left(\frac{C_{2j}^{j}}{4^j}\right)^2 \underset{n\to\infty}{\sim} \dfrac{1}{j\pi}$ and finally $\mathbb{E}_0[N_0] = \infty$

Proposition 12. *The symmetric two dimensional random walk is null recurrent.*

The reccurence is proved by the assumptions above and the nullity results from the nullity in dimension 1

Complexity for Algorithm 1 Here, as in dimension 1 the convergence time is infinite:

Proposition 13. *The convergence time of Algorithm 1 in a 2-dimension unbounded environment is infinite.*

Remark 14. In dimension 3 or higher the symmetric random walk is transient, so the petrels can be lost forever, algorithm 1 does not converge in dimension 3.

Here also, to reduce the convergence time we delimit the environment by the introduction of reflecting barriers.

Bounded Environment. The petrels evolves in this section as a random walk in the plan with reflective barriers at the borders which is described in section 3.2. Let $(X_n)_{N\in\mathbb{N}}$ be the random variable which represents the position of the petrel in the plan at the moment n. One can also write $X_n = (x_n, y_n)$ with $(x_n)_{N\in\mathbb{N}}$ and $(y_n)_{N\in\mathbb{N}}$ independent.
Indeed, At every moment it draws a direction according to the x-axis (OX) and one direction according to the y-axis (OY).

We define which represents the number of returns to the antenna before time n: $U_n = \sum_{i=1}^{n} \mathbb{1}_{\{X_i=(0,0)\}}$

But, $\sum_{i=1}^{n} \mathbb{1}_{\{X_i=(0,0)\}} = \sum_{i=1}^{n} \mathbb{1}_{\{x_i=0\}\cap\{y_i=0\}}$
The variables are independent so we get to
$\mathbb{E}_0[U_n] = \sum_{i=1}^{n} E[\mathbb{1}_{\{X_i=(0,0)\}}] = \sum_{i=1}^{n} E[\mathbb{1}_{\{x_i=0)\}\cap\{y_i=0)\}}] = \sum_{i=1}^{n} E[\mathbb{1}_{\{x_i=0)\}}]E[\mathbb{1}_{\{y_i=0)\}}]$
According to what we saw in dimension 1 we knows that $\mathbb{E}_0[\mathbb{1}_{\{x_i=0)\}}] = \mathbb{P}(x_i = 0) \to_{n\to\infty} \frac{1}{2(N-1)}$ and $\mathbb{E}_0[\mathbb{1}_{\{y_i=0)\}}] = \mathbb{P}(y_i = 0) \to_{n\to\infty} \frac{1}{2(N-1)}$ We obtain a divergent sequence, therefore partials sums are equivalent :

Proposition 15. *The number of returns to the antenna before n is given by* $\mathbb{E}_0[U_n] \sim_{n\to\infty} \frac{n}{4(N-1)^2}$

Complexity for Algorithm 1

Proposition 16. *The convergence time for algorithm 1 in a 2-dimension environment with reflecting barriers is $O(4N^2P)$*

Proof:
As in the previous paragraph the result is given by the number of return before n.

5 The To-Antenna-and-To-Petrels (TATP) Model

In the TATP model petrels can communicate one with another. To compute the convergence time for the algorithms related to this model, we need to evaluate the time needed for two random walk to meet and the number of meeting we can hope in a certain amount of time.

5.1 The TATP in Dimension 1

To model their meeting we will study the random variable which value is the difference between the two random walks : i.e: $\forall n \in \mathbb{N}$ $Z_n = X_n - Y_n$.

Lemma 17. – *$\{Z_n\}_{n\in\mathbb{N}}$ is a random walk*
– *The two random walks,$\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$, meet when $Z_n = 0$*

We will study the return to 0 of this new random walk.
Using the fact that$\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are independent we easily get the distribution of $\{Z_n\}_{n\in\mathbb{N}}$:

- $\mathbb{P}[Z_n = +2] = \mathbb{P}[X_n = +1] + \mathbb{P}[Y_n = +1] = \frac{1}{4}$
- $\mathbb{P}[Z_n = 0] = \mathbb{P}[\{X_n = +1\} \cap \{Y_n = -1\}] + \mathbb{P}[\{X_n = -1\} \cap \{Y_n = +1\}] = \frac{1}{2}$
- $\mathbb{P}[Z_n = -2] = \mathbb{P}[X_n = -1] + \mathbb{P}[Y_n = -1] = \frac{1}{4}$

Unbounded Environment .
We consider that petrels evolve like described in figure in section 3.2. In this section, we won't give any details. Indeed, even if it is not exactly the random walk we met before, the reasoning is quite identical. We can show that this random walk is null-recurrent and therefore

$$\mathbb{E}[N_0] = \infty$$

Complexity for Algorithm 2 and 3 The fact that petrels can meet one with another is not sufficient to ensure a finite expected convergence time in an unbounded environment. Therefore:

Proposition 18. *The convergence time for Algorithms* 2 *ans* 3 *is infinite in an unbounded environment.*

The adding of reflecting barriers allows to reduce the convergence time in the TA model, we will use the same technique in the TATP model.

Bounded Environment. In this section the petrels evolves on a segment. To analyze the number of meetings we have again to study the distance between the two random walks. The presence of barriers introduce a new difficulty. The difference is not a random walk anymore (not in 1 dimension). Indeed, if the distance between the two random walks is k the next distance is not determined, it depends whether or not a petrel is on a bound.
To overcome this problem, we choose to have a lower bound of the number of meetings of the two random walks by the number of return to 0 to a simplest random walk with barriers only if the distance is equal to n:
Here is the fundamental theorem of the section.

Theorem 19. *Consider $\{X_n\}_{n\in\mathbb{N}}$ an HMC with reflecting barriers at* 0 *and N.The state space of such a chain is $E = \{0, 1, \dots, N\}$.*
We define the following transition matrix.

$$\begin{pmatrix}
0 & 1 & & & & & & & \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & & & \\
& \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & & \\
& & \ddots & \ddots & \ddots & & & & \\
& & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & \\
& & & & \ddots & \ddots & \ddots & & \\
& & & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
& & & & & & & 1 & 0
\end{pmatrix}$$

We have : $\mu(0) = \frac{1}{1+2N}$

Proof. A sketch for the proof may be found in [5].

Using corollary 7, we have an estimation of the number of meetings of two random walks: $\mathbb{E}_0[\sum_{k=1}^{n} \mathbb{1}_{\{Z_k=0\}}] \underset{n\to\infty}{\sim} \frac{n}{1+2N}$

We just have to adapt this theorem to our random walk. The main difference is that $\{Z_n\}_{n\in\mathbb{N}}$ evolves 1 step by 1 step. But in our model, the walk advance of 2 into 2. This fact is not a problem, as it will only modify the state space we work on. Indeed, a solution is to consider two different kind of states : the odd and the even.

Complexity for Algorithm 2 and 3

Proposition 20. *In this environment, the convergence time is $O(6N)$ for algorithm 3 and $O(2NP)$ for algorithm 2*

Proof:
The proof is, here again, an easy consequence of the number of returns to the antenna.

5.2 The TATP in Dimension 2

Unbounded Environment . Again, we don't develop this part as it still the same, the random walk is null recurrent and $\mathbb{E}[N_0] = \infty$

Bounded Environment . Consider 2 petrels evolving like the random walk described in section 3.2. We note X_n and Y_n their positions in the plan and we denote by Z_n their difference. The variable Z_n follows a random walk with 9 possible movements of longer 2 :

- No move with probability 1/4.
- North, South, East or West with probability 1/8 each.
- NE, NW, SE, or SW with probability 1/16 each.

The key point is that movement is exactly the obtained by choosing independently the horizontal moves and the vertical moves following the random walk of the previous section.
Indeed, the method used in the analyze on the TA model can be used here.

Complexity for Algorithm 2 and 3 We, thus obtain:

Proposition 21. *The convergence time for algorithm 2 in a 2-dimension environment with reflecting barriers is $O(4N^2P)$*
The convergence time for algorithm 3 in a 2-dimension environment with reflecting barriers is $O(12N^2)$

6 Final Remarks

6.1 Application to others population protocols based algorithms

In [4], [3], the notion of rounds is replaced by the notion of interactions. But it is more or less the same. In these papers, the complexity of the algorithms are $O(P^2 lnP)$ interactions. More realistic bounds may be $O(N^d P^2 lnP)$ where N is the size of the environment, d is the dimension where evolves the sensors, if we assume that the sensors move as a random walk.

6.2 Conclusion

In this paper, we studied the complexity of algorithms based on population protocols. Our principal goal is to implement them on mobile sensor networks. We notice here that the efficiency of these kind of algorithms is strongly dependent on the size of the topology. The fact that the battery life of the sensors is small forced us to begin our implementation on small size environment. Moreover, one can argue that the type of communication (message reading) is not very realistic for sensor networks, it is why we are now working on new algorithms using message transfers with possible loss.

7 References

[1] R. Muntz A. Chen and M. Srivastava. An analysis of a large scale habitat monitoring application. In chapter in Smart Environments, editor, *Smart Rooms*. Technology, Protocols and Applications, Diane Cook, Sajal Das, eds. Wiley-Interscience, September 17, 2004.

[2] D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. In *Proc. of 23 Annual Symposium on Principle of Distributed Computing PODC 2004*, 2004.

[3] D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. In LNCS, editor, *Proc. of 20th International Symposium on Distributed Computing DISC 2006*, pages 61–75, 2006.

[4] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. In *Proc. of 25 Annual Symposium on Principle of Distributed Computing PODC 2006*, 2006.

[5] P. Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, 1999.

[6] L. de Alfaro. *Formal Verification of Probabilistic systems*. PhD Thesis, Stanford University, 1997.

[7] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, and E. Ruppert. When birds die: Making population protocols fault-tolerant. In *DCOSS*, pages 51–66, 2006.

[8] L. Fribourg, S. Messika, and C. Picaronny. Coupling and Self-stabilization. In *18th International Conference on Distributed Computing (DISC'04), Springer-Verlag, LNCS 3274*, pages 201–215. Springer, 2004.

[9] S.Messika L. Rosaz J.Beauquier, J. Clement and B. Rozoy. Self-stabilizing counting in mobile sensor networks. In *Technical Report L.R.I , number 1470*, March 2007.

[10] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring, 2002.

[11] G. Norman. Analysing randomized distributed algorithms. In *In Validation of Stochastic Systems: A Guide to Current Research*, page volume 2925. Lecture Notes in Computer Science, August 2004.

[12] J. Ouvrard. *Probabilités*. Cassini, 2000.

[13] Y. Wang M. Martonosi L. Peh P.Juang, H. Oki and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS-X conference*, October, 2002.

[14] A. Mainwaring J. Anderson R. Szewczyk, J. Polastre and D. Culler. An analysis of a large scale habitat monitoring application. In *The Second ACM Conference on Embedded Networked Sensor Systems*. SenSys 2004: 214-226, Nov. 2004.

Appendix 1: Notations

X_k is called a random variable. In our case it can be the position of a petrel after k steps of the algorithm.
We will study a discrete-time stochastic process, that is : a sequence $\{X_n\}_{n\geq 0}$ of random variables.
We will use the probability theory in order to measure the likeliness of such events occurrence.

- We note $\mathbb{P}[X_n = x]$ the probability of the event $\{X_n = x\}$.
- We note $\mathbb{E}[X_n = x]$ the expectation of the event $\{X_n = x\}$.
- We note $\mathbb{P}_i[X_n = x]$ or $\mathbb{E}_i[X_n = x]$ when the initial value of the random variable X is i.

Working with random variables will lead us to study their *probability distribution*. Indeed, every random variable gives rise to a probability distribution. This concept is quite important as this distribution contains most of the useful information about the variable. The probability distribution of a random variable X is :
$\forall\ k \quad \mathbb{P}[X = k]$
Finally we will use conditional probability, which is the probability of some event A, given the occurrence of some other event B.
Conditional probability will be written $\mathbb{P}[A/B]$, and will be read "the probability of A, given B".

Example 22 (Toy example). In our toy example we consider X_n as the position of the mobile agent after n steps, we have:
$\mathbb{P}[X_{n+1} = 1/X_n = 0] = 1$
$\mathbb{P}[X_{n+1} = N - 1/X_n = N] = 1$
$\mathbb{P}[X_{n+1} = k + 1/X_n = k] = \frac{1}{2}$
$\mathbb{P}[X_{n+1} = k - 1/X_n = k] = \frac{1}{2}$

Appendix 2: Definitions needed for stability theory

First, we define the transition matrix $P = \{p_{ij}\}_{i,j \in E}$ where

$$p_{ij} = \mathbb{P}\left[X_{n+1} = j / X_n = i\right] \tag{2}$$

as the transition matrix of the HMC.

Example 23 (Toy example). In our symmetric random walk the matrix is given by:

$$\begin{pmatrix}
0 & 1 & & & & & & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & & & & & & & \\
& \frac{1}{2} & 0 & \frac{1}{2} & & & & & & & \\
& & \cdot & \cdot & \cdot & & & & & & \\
& & & \cdot & \cdot & \cdot & & & & & \\
& & & & \cdot & \cdot & \cdot & & & & \\
& & & & & \frac{1}{2} & 0 & \frac{1}{2} & & & \\
& & & & & & \cdot & \cdot & \cdot & & \\
& & & & & & & \cdot & \cdot & \cdot & \\
& & & & & & & & \cdot & \cdot & \cdot \\
& & & & & & & & \frac{1}{2} & 0 & \frac{1}{2} \\
& & & & & & & & & 1 & 0
\end{pmatrix}$$

Definition 24 (Stationary distribution). A probability distribution μ satisfying

$$\mu^t = \mu^t P \tag{3}$$

where μ^t is the transposition of μ, is called a stationary distribution of the transition matrix P, or of the corresponding homogeneous Markov chain.

We easily see that if the stationary distribution of the homogeneous Markov chain is its initial distribution, it keeps the same distribution to infinity. What we want to know is how will converge the long-run behavior of the homogeneous Markov chain if we choose an arbitrary initial distribution. If we can answer to this question, it will allow us to describe the evolution of our model.

Definition 25. State j is said accessible from state i if there exists a non null probability to go from state j to state i. In other words, if there exists a path to go from j to i. In the same way, state i and j are said to communicate if i is accessible from j and j is accessible from i.

We are now able to define communication classes. Indeed the communication relation is an equivalence relation and generates a partition of the space E into equivalent classes called communication classes. Furthermore we say that two classes communicate if there exists a state in each classes that communicate.

Definition 26. If there exists only one communication class, then the chain is said to be irreducible

Example 27 (Toy example). The symmetric random walk on a segment is irreducible, it is obvious that every state communicate with another one by choosing the good probabilistic transitions.

In order to understand how the homogeneous Markov chain will evolve we want to answer the following questions :

- Will all states be visited ?
- Do all classes communicate ?

That is the purpose of the following formal definition :

Definition 28. We define $T_i = \inf\{n \geq 1; X_n = i\}$ the return time to state $i \in E$. State i is called positive recurrent if

$$\begin{cases} \mathbb{P}_i(T_i < \infty) = 1 \\ \mathbb{E}_i[T_i] < \infty \end{cases} \text{ Otherwise, it is called null recurrent}$$

Intuitively a state is said to be recurrent if we are almost sure it will be visited, ans it said positive recurrent if moreover the expectation of the time of return is finite. All theses definitions can be extend to a HMC. For example a HMC is said to be recurrent if and only if all his states are recurrent. We can note that a recurrent HMS with finite state space is always positive.

Example 29 (Toy example). The random walk on a segment is recurrent positive. The proof is quite simple: all states are in the same communication class, and therefore, are all recurrents; the state space is finite so the chain is positive recurrent.

Appendix 3: Proof of the null-recurrence.

Is it recurrent ? .
We define : $O_j = \mathbb{1}_{\{X_j=0\}}$. This leads too

$$\begin{cases} \mathbb{P}_0(O_{2j+1} = 1) = 0 \\ \mathbb{P}_0(O_{2j} = 1) = \frac{C_{2j}^j}{4^j} \end{cases}$$

Indeed, to be at the origin at the moment $2j$ means that the petrel did exactly j steps on the left and j on the right. The probability of such an event is well such as above.
Therefore :

$$\mathbb{E}_0[N_0] = \mathbb{E}_0[\sum_{n\geq 1} \mathbb{1}_{\{X_n=0\}}] = \sum_{n\geq 1} \mathbb{P}_0[\mathbb{1}_{\{X_n=0\}}] = \sum_{n\geq 1} \frac{C_{2j}^j}{4^j}$$

But we knows that

$$\frac{C_{2j}^j}{4^j} \underset{n\to\infty}{\sim} \frac{1}{\sqrt{j\pi}}$$

Therefore, the sequence is divergent and we have $\mathbb{E}_0[N_0] = \infty$. The symmetric random walk is recurrent.

Is it recurrent positive ? .

Let $\tau_1 = T_0, \tau_2, ...$ be the successive return times to state 0 of the random walk on $\mathbb{Z}$. First we recall that according to what we just saw :

Lemma 30. $\mathbb{P}_0[T_0 < \infty] = 1$

Now, we can observe that for $n \geq 1$,

$$\mathbb{P}_0[X_{2n} = 0] = \sum_{k\geq 1} \mathbb{P}_0[\tau_k = 2n]$$

and therefore, for all $z \in \mathcal{C}$ such that $|z| < 1$,

$$\sum_{k\geq 1} \mathbb{P}_0[X_{2n} = 0]z^{2n} = \sum_{k\geq 1}\sum_{n\geq 1} \mathbb{P}_0[\tau_k = 2n]z^{2n} = \sum_{k\geq 1} \mathbb{E}[z^{\tau_k}]$$

We are now going to use the following proposition

Proposition 31. *Let $\{X_n\}_{n\geq 0}$ be an HMC with initial state 0 that is almost surely visited infinitely often. Denoting $\tau_1 = T_0, \tau_2, ...$ be the successive return times to state 0,*
the pieces of trajectory $\{X_{\tau_k}, \{X_{\tau_k+1}, \ldots, \{X_{\tau_{k+1}-1}\}, k \geq 0$ are independent and identically distributed.

Proof. We do not give the detail of the proof which is just a little manipulation of random variables.

Therefore, noticing that $\tau_k = \tau_1 + (\tau_2 - \tau_1) + \ldots + (\tau_k - \tau_{k-1})$ and $\tau_1 = T_0$, we get

$$\mathbb{E}_0[z^{\tau_k}] = (E_0[z^{T_0}])^k$$

In particular,

$$\sum_{n\geq 0} \mathbb{P}_0[X_{2n} = 0]z^{2n} = \frac{1}{1 - \mathbb{E}_0[z^{T_0}]}$$

So, after the evaluation of the left hand side :

$$\sum_{n\geq 0} \frac{1}{2^{2n}} \frac{(2n)!}{n!n!} z^{2n} = \frac{1}{1-z^2}$$

Therefore the generating function of the return time to 0 given $X_0 = 0$ is $\mathbb{E}_0[z^{T_0}] = 1 - \sqrt{1-z^2}$. Its first derivative

$$\frac{z}{\sqrt{1-z^2}}$$

tends to 0 as $z \to 1$. Therefore using Abel's theorem

$$\mathbb{E}_0[T_0] = \infty$$

We see that although given $X_0 = 0$ the return time is almost surely finite, it has an infinite expectation.