Merging Time of Random Mobile Agents

Shehla Abbas, Mohamed Mosbah, Akka Zemmari
LaBRI, Université Bordeaux 1
351 cours de la liberation,
33405 Talence-France
{abbas,mosbah,zemmari}@labri.fr

Abstract. In this paper, we investigate the following problem: k mobile agents are placed on a graph G. Each agent performs a random walk, and when two or more agents meet at a node, they merge into a single agent. We present a Markov chain modeling the agents behaviour and show how this can be used to upper bound the expected time for all the k agents to merge into a single agent.

Keywords: Mobile Agents, Random Walks, Rendezvous, Meeting Time.

1 Introduction

In today's world, immense importance has been given to distributed computing and algorithms to give the maximum throughput with minimum resources. Mobile agents are one of the ways to implement these computations and algorithms. In recent years emphasis has also been shifting from deterministic algorithms to randomized algorithms which are often more simple and more efficient. So we are also interested in randomized algorithms and solutions for the displacements of agents in the network. These algorithms require less energy and few resources to perform any delegated task.

The Model. We consider a *synchronous distributed* network of processors with an arbitrary topology. It is represented as a connected graph where vertices represent processors, and two vertices are connected by an edge if the corresponding processors have a direct communication link.

A *mobile agent* (MA) is an entity which moves in the graph. There are many models to study the movement of the agents depending on the parameters of interest and on the applications. In this paper, we consider MAs with identities whose storage memory can not deal with topological information of the network. This is useful for instance if the network is huge(e.g. Internet). We study in this work agents that do not store information about the visited vertices. One interesting model that we deal with is that of random walks. Hence, our MA moves randomly in the graph. The MA performs a *random walk*: when it is located in a vertex v, if $B(v, 1)$ denotes the ball of center v and radius 1, the MA chooses *uniformly at random* one of the vertices in $B(v, 1)$, if the chosen vertex is v, the MA does not move, else, it moves to the chosen vertex.

The Problem. In this paper, we are interested in the following problem: a set of $k \geq 2$ mobile agents are placed arbitrarly on the network. These agents can work in collaboration by performing random walk and can give efficient results by the combined effort (for example in applications such as collecting information, spanning tree or gathering problem), however, the agents are reduced each time the agents meet on any node v of the netowrk. Upon meeting the agents can share their information they already acquired by doing some computations. In this case, the agent with the smaller identity disappers itself from the nework. At the end of our algorithm all agents merge into a *single* MA having the biggest identity.

Another objective of this paper is to calculate the expected time for all the agents to meet in a network hence analyzing the complexity of the algorithm. We make another assumption that a MA remains on the vertex where it is located or moves to one of the vertices in the neighbourhood. Indeed, without this hypothesis, even if we use random algorithm to break symmetry, the *rendezvous* can never occur. To see that, consider the case of the complete graph K_2 with two vertices and two MAs one on each vertex. Then if we do not allow the MAs to remain on a vertex, they will never meet.

Related Works. The calculation of the time for two agents to meet in a graph can be considered as the *rendezvous problem*. A lot of work has been done on the rendezvous problem and vast literature can be found in [6]. Rendezvous on graphs has been presented in [3,4]. Some papers also discuss the probabilistic solutions like in [4,5,7,9]. Gathering is another technique [11,14,15] when more than two agents have to meet at one fixed location. Random walk is a useful tool for many applications such as the construction of a spanning tree by multiple mobile agents [2]. It has many other applications and it can also be considered as an Election Problem [8]. In another work of agent rendezvous by [16], the authors considered their agents (or distributed units as they call it) to be intelligent agents having computing power, control and memory that is having decision capability which helps in understanding the topology of the network. The agents are coordinated of their behaviour in a distributed environment. The meeting time of agents for the complete graph is $O((n/k)\log n)$ with high probability and for hypercube, it is $O((n/k)\log^2 n)$ with high probility, however their model is different from ours in which agents do not have control, intelligence and memory. A survey on mobile agent rendezvous can be found in [13]. In this survey, authors have narrated various results on the mobile agent rendezvous on distributed networks having deterministic and randomized rendezvous among agents; the agents having computing power, control and memory and the number of agents vary from *2* to *k*. To the best of our knowledge, the present paper is the first to consider the *randomized* meeting of any k agents without memory and the reduction of the number of agents to one.

Our Contribution. In this paper, we first give a Markov chain formulation of the problem, then we investigate some particular cases and analyze the merging time in complete graphs, star graphs and the hypercube.

1.1 An Introductory Example

As a first investigation, consider a graph $G = (V, E)$ with a star topology. Consider two agents in this graph. Depending on the placement of the two agents, if we consider the random variable $(X_t)_{t\geq 0}$, such that X_t is the distance between the two agents at time t, then $\forall t \geq 0$, $X_t \in \{0, 1, 2\}$ and its behaviour can be modelled by a Markov chain $\mathcal{M} = (\{0, 1, 2\}, \mathcal{T})$, where $\mathcal{T}$ is defined by the following transition matrix:

$$\mathcal{P} = \left(\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 0 & 0 \\ 1 & \frac{1}{n} & \frac{1}{2} & \frac{n-2}{2n} \\ 2 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array}\right)$$

This chain is an absorbing one, the single absorbing state is 0 and its *fundamental matrix* (see [12]), can be written as follows

$$\mathcal{N} = \left(\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & \frac{6n}{n+4} & \frac{4(n-2)}{n+4} \\ 2 & \frac{4n}{(n+4)} & \frac{4n}{n+4} \end{array}\right)$$

Hence, if we denote by T_i the time for the two agents to obtain a rendezvous, given that they are initially placed at distance i ($i \in \{1, 2\}$) one from the other, we have

$$\mathbb{E}(T_1) = \frac{10n - 8}{n + 4}$$

and

$$\mathbb{E}(T_2) = \frac{8n}{n + 4}.$$

Hence, if we denote by T the time for the two agents to obtain a rendezvous, we have, $\mathbb{E}(T) = O(1)$.

If we consider $k \geq 3$ MAs in the star graph, the parameter of interest will be the number of MAs in the graph and the behaviour of this number. Then, instead of the distance between MAs, we consider the r.v. $(X_t)_{t\geq 0}$ which counts the number of MAs in the network at time t. We then start by the following lemma:

Lemma 1. *Let X_t be the number of agents on the leaves at time $t \geq 0$. Then*

$$\mathbb{E}\left(X_{t+1} \mid X_t\right) \leq \frac{X_t}{2} + 1.$$

Proof. To prove the lemma, we have to consider two cases:

1. There are 0 MAs at the center. Let $\mathbb{E}_0(X_{t+1} \mid X_t)$ denotes the expected number of agents on the leaves at time $t + 1$ knowing that there were X_t agents on the leaves at time t, then, for any $i \in \{0, 1, \cdots, X_t\}$, if i MAs move to the center of the star, then the number of MAs on the leaves decrease by i, hence, a combinatorial and probabilistic reasoning yields to

$$\mathbb{E}_0(X_{t+1} \mid X_t) = \sum_{i=0}^{X_t} \left(\frac{1}{2}\right)^{X_t} \binom{X_t}{i} (X_t - i) = \frac{X_t}{2}$$

2. There is an MA at the center. Let $\mathbb{E}_1(X_{t+1} \mid X_t)$ denotes the expected number of agents on the leaves, a similar reasoning as in 1) yields to

$$\mathbb{E}_1(X_{t+1} \mid X_t) = \left(1 - \frac{1}{n}\right) \frac{X_t}{2} + \left(1 - \frac{1}{n}\right).$$

The lemma follows. □

Then we have the following theorem.

Theorem 1. *If we denote by T_k the time for k MAs to merge into a single MA, then*

$$\mathbb{E}(T_k) = O(\log(k)).$$

Proof. From the formulae in Lemma 1, we can see that the number of times we can apply the recursion is given by $\log(X_0)$. Since $X_0 \leq k$, the theorem follows. □

2 A Genral Markov Chain Formulation

2.1 Configurations Graph

The first formulation of the problem is inspired from [10]. As in [10], a *configuration* of k agents on the graph G is a multiset of k vertices of G. A configuration $\{x_1, x_2, \cdots, x_k\}$ describes a placement of k agents on the vertices of G, where x_i is the *position* of agent a_i. If $x_i = x_j$ then the agents a_i and a_j are merged into one agent: a_i if $a_i > a_j$ and a_j otherwise. The process starts in an *initial configuration* $\nu_0 = \{x_1, x_2, \cdots, x_k\}$ with $x_i \neq x_j, \forall i \neq j$ and ends with a *terminal configuration* $\{x_1, x_2, \cdots, x_k\}$ with $x_i = x_j, \forall i \neq j$. If $\nu_p = \{x_1, x_2, \cdots, x_k\}$ is the configuration of the process at step p then a *next configuration* $\nu_{p+1} = \{y_1, y_2, \cdots, y_k\}$ of ν_p satisfies the following conditions:

1. if $y_i \neq x_i$ then $y_i \in N(x_i)$ where $N(x_i)$ denotes the set of neighbours of x_i,
2. for all $i \neq j$, if $x_i = x_j$ then $y_i = y_j$

An *execution* starting at initial configuration ν_0 is a finite sequence of configurations $\{\nu_0, \nu_1, \cdots, \nu_m\}$ such that for all $p = 0, 1 \cdots, m-1$, ν_{i+1} is a next configuration of ν_i. Configuration ν_m is called the *current configuration* at time step m. $\nu_m = \{x_1, x_2, \cdots, x_k\}$ is a *terminal* configuration if $x_i = x_j$ for all $i \neq j$, this confuguration is noted $TERM$.

As in [10], the *configuration graph* $G_k = (V(G_k), E(G_k))$ is the directed graph defined as follows: $V(G_k)$ is the set of all the possible configurations and

$$E(G_k) = \{(\nu_1, \nu_2) \mid \nu_1 = \{x_1, x_2, \cdots, x_k\} \neq TERM \text{ and}$$

$$\nu_2 = \{y_1, y_2, \cdots, y_k\} \text{ is a next configuration of } \nu_1\} \bigcup \{(TERM, TERM)\}$$

Since G is connected, it is clear that G_k is connected. The behaviour of the set of agents $\{a_1, a_2, \cdots, a_k\}$ in the undirected graph G is then the same as the behaviour of a random walk on the directed configuration graph G_k. This random walk is a Markov chain with one absorbing state $TERM$, the other states are all transient.

Since G_k is connected, it is clear that, with probabilty 1, the random walk in G_k will reach the state $TERM$, meaning that all the MAs will end by merging on a single MA.

In this study, we are interested on the expected time before all the agents merge into one agent. This can be interpreted as the expected time for a random walk starting in an initial configuration to reach the state $TERM$ in G_k. In terms of random walks, this corresponds to the notion of *hitting time* $H(\nu_0, TERM)$.

2.2 Components Graph

Let G be a connected graph. As we have seen in the previous section, the behaviour of k MAs on G is the same as the bahaviour of a random walk on the configurations graph G_k. Nevertheless, this random walk, even if it is a Markov chain, seems not easy to study and do not provide a simple method to upper bound the merging time of k MAs on G.

Let ν_i be any configuration, we define the *size* of ν_i, and we denote it $\mid \nu_i \mid$, the number of different vertices in ν_i. We define the *components graph* $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ as follows:

- $\mathcal{V}_k = \{C_1, C_2, \cdots, C_k\}$ such that $\forall i \in \{1, 2, \cdots, k\}$, $C_i = \{\nu \in V_k \mid |\nu| = i\}$
- $\mathcal{E}_k = \{(C_i, C_j) \mid \forall \nu_1 \in C_i, \exists \nu_2 \in C_j \text{ such that } (\nu_1, \nu_2) \in E_k\}$

The main idea of this construction is to gather, on a single component, all the configurations of the same size and there is a transition from a component C to a component C' iff for any configuration ν_1 in C, there is at least a configuration ν_2 in C' such that $(\nu_1, \nu_2) \in E_k$.

Let $(Y_t)_{t\geq 0}$ be a random walk on G_k. For any $(i,j) \in \{1,2,\cdots,k\} \times \{1,2,\cdots,k\}$, we define $p_{i,j}$ as follows:

$$p_{i,j} = \begin{cases} \min\{\mathbb{P}r\left(Y_{t+1} = \nu_2 \mid Y_t = \nu_1\right) \mid (\nu_1,\nu_2) \in C_i \times C_j\} & \text{if } i > j \\ 1 - \sum_{l<i} p_{i,l} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Then, consider the following process: $X_0 = C_k$ and for any $t \geq 0$,

$$\mathbb{P}r\left(X_{t+1} = C_j \mid X_t = C_i\right) = \begin{cases} p_{i,j} & \text{if } i > j \\ 1 - \sum_{l<i} p_{i,l} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

We have the following lemma:

Lemma 2. *If $\mathcal{G}_k$ is connected then*

$$H(\nu_0, TERM) \leq H(C_k, C_1), \text{ for any initial configuration } \nu_0.$$

Proof. By induction on k. □

2.3 From 2 to k Agents

In this section we show how the study of the particular case of two agents in a graph $G = (V,E)$ allows us to obtain an upper bound for the general case of $k \geq 3$ MAs in G.

Let $G = (V,E)$ be a given graph of size n and $k \geq 3$ be the number of the MAs in G. Let $\mathcal{G}_k$ the corresponding components graph. In this section, we assume that $\mathcal{G}_k$ is connected. Consider a random walk in $\mathcal{G}_k$, and let $(X_t)_{t\geq 1}$ be the r.v. defined as follows: $X_0 = k$ and $\forall t > 0$, X_t is the size of the component at time t.

The behaviour of the r.v. $(X_t)_{t\geq 0}$ can be modelled by a Markov chain $\mathcal{M} = (\mathcal{S}, \mathcal{T})$ such that $\mathcal{S} = \{1,\cdots,k\}$ and

$$\mathbb{P}r\left(X_{t+1} = 1 \mid X_t = 1\right) = 1$$
$$\mathbb{P}r\left(X_{t+1} = j \mid X_t = i\right) = \begin{cases} p_{i,j} > 0 & \text{if } j \leq i \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{M}$ is an absorbing Markov chain with a single absorbing state (the state 1) with the following transition matrix:

$$P = \left(\begin{array}{c|cccccc} & 0 & 1 & 2 & \cdots & k-1 & k \\ \hline 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & p_{2,1} & p_{2,2} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \cdots & \cdots \\ k & p_{k,1} & p_{k,2} & p_{k,3} & \cdots & p_{k,k-1} & p_{k,k} \end{array}\right)$$

Let $(Y_t)_{t\geq 0}$ be the r.v. defined as follows: $Y_t \in \mathcal{S}$, $\forall t \geq 0$ and

$$\mathbb{P}r\left(Y_{t+1} = 1 \mid Y_t = 1\right) = 1$$
$$\mathbb{P}r\left(Y_{t+1} = j \mid Y_t = i\right) = \begin{cases} q_{i,j} = p_{i,j} & \text{if } j = i \\ q_{i,j} = \sum_{l=i-1}^{1} p_{i,l} & \text{if } j = i-1 \\ 0 & \text{otherwise.} \end{cases}$$

This corresponds to an absorbing Markov chain $\mathcal{M}' = (\mathcal{S}, \mathcal{T}')$ with the following transition matrix:

$$P' = \left(\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & \cdots & k-1 & k \\ \hline 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 2 & q_{2,1} & q_{2,2} & 0 & 0 & \cdots & 0 & 0 \\ 3 & 0 & q_{3,2} & q_{3,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots \\ k & 0 & 0 & 0 & \cdots & 0 & q_{k,k-1} & q_{k,k} \end{array}\right)$$

$\mathcal{M}'$ is a Markov chain whose set of states is the same as for $\mathcal{M}$ and the only possible transitions are between i and $i-1$. This intuitively means that whereas in $\mathcal{M}$, from a state i it is possible to reach all the states $i-1, i-2, \cdots, 1$, one can reach only the state $i-1$ in $\mathcal{M}'$. It is clear that :

Lemma 3. *For any $i \in \{1, 2, \cdots k\}$, if we denote by T_i (resp. T_i') the time for the Markov chain $\mathcal{M}$ (resp. $\mathcal{M}'$), starting at the state i to reach the state 0, then*

$$\mathbb{E}(T_i) \leq \mathbb{E}(T_i').$$

Proof. From the transition matrices P and P', we deduce the following recursions: $\forall i \geq 2$, $\mathbb{E}(T_i) = 1 + p_{i,i}\mathbb{E}(T_i) + \sum_{j=1}^{i-1} p_{i,j}\mathbb{E}(T_j)$ and $\mathbb{E}(T_i') = 1 + p_{i,i}\mathbb{E}(T_i') + \left(\sum_{j=1}^{i-1} p_{i,j}\right)\mathbb{E}(T_j)$. Hence, an induction reasoning yields to $\mathbb{E}(T_i) \leq \mathbb{E}(T_i'), \forall i \geq 2$. □

Application : The Complete Graph K_n

Let $K_n = (V, E)$ be the complete graph of size n. We start by studying the case of $k = 2$ MAs. If we denote by D the distance between the two MAs in K_n, it is clear that $D \in \{0, 1\}$. Then, if T is the r.v, which counts the number of steps before the two MAs meet, T is a r.v. with a geometric distribution of the parameter

$$p = \mathbb{P}r(D = 1) = \frac{1}{n},$$

hence, for any $m \geq 1$,

$$\mathbb{P}r(T = m) = p(1-p)^{m-1} = \frac{1}{n}\left(1 - \frac{1}{n}\right)^{m-1}.$$

Yielding to the expected number of steps before the two MAs meet:

$$\mathbb{E}(T) = \frac{1}{p} = n$$

Now, let $k \geq 3$ be the number of agents in the graph K_n. From Lemma 3, it suffices to study the Markov chain $\mathcal{M}'$. To do so, we have to calculate the transition probabilities $q_{i,i}$ and $q_{i,i-1}$, $\forall i \in \{2, 3, \cdots, k\}$.

Lemma 4. *Let $\mathcal{M}'$ be the Markov chain defined above. Then, for any $i \in \{2, 3, \cdots, k\}$, we have the following:*

$$q_{i,i} = \frac{(n-1)(n-2)\cdots(n-i)}{(n-1)^i} \tag{1}$$

and $q_{i,i-1} = 1 - q_{i,i}$.

This yields to:

Lemma 5. *If we denote by T the time for all the k MAs to merge in a single agent, then*

$$\mathbb{E}(T) = O(n \log k).$$

Proof. Once $(n-j) \leq (n-1)$, $\forall j \geq 1$, we obtain from equality (1) that $q_{i,i} \leq \frac{n-i}{n-1}$ for any $i \in \{2, 3, \cdots, k\}$. On the other side, we have

$$\mathbb{E}(T) = \mathbb{E}(T'_k) = \sum_{i=2}^{k} \frac{1}{q_{i,i-1}}.$$

Hence,

$$\mathbb{E}(T) \leq \sum_{i=2}^{k} \frac{n-1}{n-i} = (n-1)H_k,$$

where $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$ is the harmonic number. Since $H_k \sim \log(k)$ the lemma follows. □

3 Hypercubes

An hypercube of dimension d is a graph $\mathcal{H}_d = (V, E)$ such that $V = \{\omega \in \{0,1\}^d\}$ and $E = \{(\omega_1, \omega_2) \in V \times V \mid h(\omega_1, \omega_2) = 1\}$ where $\forall (\omega_1, \omega_2) \in V \times V$, $h(\omega_1, \omega_2)$ denotes the *Hamming distance* between ω_1 and ω_2, i.e., the number of bits to change in ω_1 to obtain ω_2. That is, this graph has some nice properties: each vertex has a degree d, the maximum distance between any two vertices in the graph is d and the size of the graph is $n = 2^d$.

We first investigate the case of two MAs in $\mathcal{H}_d$. As for the previous cases, we consider the random variable $(X_t)_{t \geq 0}$ such that $\forall t \geq 0$, X_t corresponds to the distance between the two MAs at time t. The behaviour of this r.v. can be modelled by a Markov chain $\mathcal{M} = (\mathcal{S}, \mathcal{T})$ where $\mathcal{S} = \{0, 1, \cdots, d\}$ and the transitions are defined as follows:

$$Pr\left(X_{t+1} = 0 \mid X_t = 0\right) = 1$$

$$Pr\left(X_{t+1} = j \mid X_t = i\right) = \begin{cases} \frac{d+2di-2i^2+1}{(d+1)^2} & \text{if } j = i \\ \frac{2i}{(d+1)^2} & \text{if } j = i-1 \\ \frac{2(d-i)}{(d+1)^2} & \text{if } j = i+1 \\ \frac{i(i-1)}{(d+1)^2} & \text{if } j = i-2 \\ \frac{(d-i)(d-i-1)}{(d+1)^2} & \text{if } j = i+2 \\ 0 & \text{otherwise.} \end{cases} \forall i \geq 0, j \geq 1,$$

That is, $\mathcal{M}$ is an absorbing Markov chain with a single absorbing state: the state 0.

Remark 1. If the distance between the two MAs is close to n, there are more chances that the distance between them decreases. If the distance between them is close to 0, which is an absorbing state, there are more chances that the distance increases. Similarly, if the distance is in the middle then there are more chances that it remains the same.

The last remark makes the study of the above chain more complicated than the previous ones. Indeed, in this case, the transition probabilities are not constant but are given by means of d and

the actual value of the distance between the two MAs. Hence, we use another method to study this process: we decompose the chain $\mathcal{M}$ into two Markov chains $\mathcal{M}_1$ and $\mathcal{M}_2$, each of them with the following transition matrix:

$$A_1 = A_2 = \left(\begin{array}{c|cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & \cdots & d \\ \hline 0 & \frac{1}{d+1} & \frac{d}{d+1} & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & \frac{1}{d+1} & \frac{1}{d+1} & \frac{d-1}{d+1} & 0 & 0 & 0 & \dots & 0 \\ 2 & 0 & \frac{2}{d+1} & \frac{1}{d+1} & \frac{d-2}{d+1} & 0 & 0 & \dots & 0 \\ 3 & 0 & 0 & \frac{3}{d+1} & \frac{1}{d+1} & \frac{d-3}{d+1} & 0 & \dots & 0 \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ d & 0 & 0 & 0 & 0 & 0 & \dots & \frac{d}{d+1} & \frac{1}{d+1} \end{array}\right)$$

If we denote by $Q = (q_{i,j})_{0 \leq i,j \leq d}$ the matrix $Q = A_1 \times A_2$ and by P the transition matrix of the chain $\mathcal{M}$ then it is easy to verify that

$$p_{i,j} = q_{i,j},\ \forall (i,j) \in \{1, 2, \cdots, d\} \times \{1, 2, \cdots, d\}. \tag{2}$$

The formulae (2) can be interpreted as follows. There is a transition, in the process modelled by $\mathcal{M}$, from a state i to a state j if and only if there is a transition, in the process modelled by $\mathcal{M}_1$, from i to a state $l \in \{0, 1, 2, \cdots, d\}$ and a transition, in the process modelled by $\mathcal{M}_2$, from l to j.

Then, rather than studying the chain $\mathcal{M}$, we first investigate the chain $\mathcal{M}_1$. The parameter of interest will be the expected value of the r.v. which counts the number of transitions for the process modelled by $\mathcal{M}_1$ to reach the state 0. Indeed, when this occurs, this means that the two MAs are at distance at most 1.

$\mathcal{M}_1$ is a regular Markov chain which admits a stationary distribution $\pi = \pi_{i(0 \leq i \leq d)}$, that is, a vector of values in $[0, 1]$ such that $\pi A_1 = \pi$ and $\sum_{i=1}^{d} \pi_i = 1$. This yields to a system which we can solve to obtain

$$\forall i \in [0, d],\ \pi_i = \frac{\binom{d}{i}}{n}.$$

Let T_i denote the time for the process $\mathcal{M}_1$ being at the state i to reach the state 0, $(T_i)_{1 \leq i \leq d}$. The expected value of T_i verifies the following system:

$$\begin{cases} \mathbb{E}(T_0) & = \frac{1}{\pi_0} = n \\ \mathbb{E}(T_{i+1}) - \mathbb{E}(T_i) & \leq \frac{1}{d-i} n,\ \forall 1 \leq i < d \end{cases}$$

Summing the left and the right sides, we obtain

$$\begin{aligned} \mathbb{E}(T_i) - \mathbb{E}(T_0) &\leq \left(\sum_{j=0}^{i} \frac{1}{d-j}\right) n \\ &\leq \left(\sum_{j=0}^{i} \frac{1}{i-j}\right) n \\ &= H_i n, \end{aligned}$$

where for any $i \geq 1$, H_i is the harmonic number. Hence,

$$\mathbb{E}(T_i) \leq H_i n + n.$$

Yielding to

$$\mathbb{E}(T_i) \sim n \log i.$$

Since $i \leq d$, we obtain the fundamental result:

Theorem 2. *With 2 MAs in the hypercube $H_d = (V, E)$, if we denote by T the time for the two MAs to obtain a rendezvous, then*

$$\mathbb{E}(T) = O(n \log\log n).$$

This permits to obtain the general result:

Theorem 3. *With k MAs in an hypercube $\mathcal{H}_d$, if T denotes the time for* all *the k MAs to merge in one agent, then*

$$\mathbb{E}(T) = O(kn \log\log n).$$

Proof. To upper bound $\mathbb{E}(T)$, we use the Lemma 3. A probabilistic reasoning yields to the fact that $q_{i,i-1} \geq q_{2,1}$, $\forall i \geq 2$, which allows us to obtain an upper bound for $\mathbb{E}(T')$:

$$\mathbb{E}(T') \leq \sum_{i=2} k \frac{1}{q_{2,1}},$$

using the result of Theorem 2, we obtain $\mathbb{E}(T) = O(kn \log\log n)$. □

Remark 2. In the introduction we said that this problem models the election problem in agents based system. This can also be used to solve a more general problem: the election problem in distributed system. Indeed, it suffices that each vertex in the graph generates a MA with the vertex identity and then the vertex which has generated the MA which is eventually elected in the n agents is the elected vertex. The complexity of the resulting algorithm is obtained by substituting n to k in the formulae.

4 Conclusion and Perspectives

In our algorithms, we calculated the time it takes for the agents to meet in a network. Starting with k agents, each one of them collect information from the nodes of the network. By working in parallel to visit nodes and by gathering their information, the global algorithm is more efficient than if it was executed by a single agent.

We calculated the complexities for different classes of graphs with two agents and then we extended it to k agents. We start our problem with star topology which takes the constant time $O(1)$ for the two agents to meet. Extending it to k agents gives the result of $O(\log k)$. In the case of a complete graph, the time it takes by two agents is linear with n. Similarly for k agents, the calculation yields $O(n \log k)$. Then we further studied the case of the hypercube; with two agents it gives the time of $O(n \log\log n)$. In case of k agents, we get the time of $O(kn \log\log n)$. This shows that it is better and more efficient than using one agent in the network.

This problem can be further extended to other kind of graphs such as d-regular graphs. We are also working on implementing these algorithms on a software tool to obtain the experimental results.

Acknowledgement. Authors are very grateful to Nasser Saheb for helpful discussion.

References

1. S. Abbas, M. Mosbah, and A. Zemmari. Collecte d'informations par des agents mobiles. In *Proceedings of New Technologies for Distributed Systems (NOTERE'2007)*. HERMES.
2. S. Abbas, M. Mosbah, and A. Zemmari. Distributed computation of a spanning tree in a dynamic graph by mobile agents. In *Proceedings of IEEE 1st International Conference on Engineering of Intelligent Systems*, pages 425-430, April 2006.

3. J. Alpern, V. Baston, and S. Essegaier. Rendezvous search on a graph. *Journal of Applied Probablity*, 3(1):223-231,1999.
4. S. Alpern. The rendezvous search problem. *SIAM J. on Control and Optimization*, 33(3)673-683,1995.
5. S. Alpern. Rendezvous search on labelled networks. *Naval Researh Logistics*, 49:256-274,2002.
6. S. Alpern and S. Gal. *The thoery of search games and rendezvous. International Series in Operations Research and Management Science, number 55.* Kluwer Academic Publishers, 2002.
7. E. Anderson and R. Weber. The rendezvous problem on discrete locations. *Journal of Applied Probablity*, 28:839-851,1990.
8. J. Bar-Ilan and D. Zernick. Random leaders and random spanning trees. In *Proceedings of the 3rd IWDAG*, 1989.
9. V. Baston and S. Gal. Rendezvous on the line when the players' initial distance is given by an unkown probability distribution. *SIAM J. on Control and Optimization*, 36:1880-1889,1998.
10. Nader H. Bshouty, Lisa Higham, and Jolanta Warpechowska-Gruca. Meeting times of random walks on graphs. *Information Processing Letters*, 69(5):259–265, 1999.
11. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous oblivious robots with limited visbility. In *Proc. 18th Annual Symposium on Theoratical Aspects of Computer Science*. LNCS 2010, pages 247-258, 2001.
12. Kemeny and Snell. *Finite Markov Chains*. D. Van Nostrand, 1960.
13. E. Kranakis, D. Krizanc, and S. Rajsbaum. Mobile agent rendezvous: A survey. In *SIROCCO*, pages 1–9. LNCS 4056, 2006.
14. W. Lim and S. Alpern. Minimax rendezvous on the line. *SIAM J. on Control and Optimization*, 34(5):1650-1665,1996.
15. L. Thomas. Finding your kids when they are lost. *Journal on Operational Reseach*, 43(5):637-639,1992.
16. X. Yu and M. Yung. Agent rendezvous: A dyanmic symmetry-breaking problem. In *23rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 610–621. LNCS 1099, 1996.