A Probabilistic Model for Distributed Merging of Mobile Agents

Shehla Abbas, Mohamed Mosbah, Akka Zemmari
LaBRI, University of Bordeaux 1,
351 cours de la Libération
33400 Talence-France
{abbas,mosbah,zemmari}@labri.fr

Abstract
We propose a probabilistic algorithm for mobile agents which roam around in the network following a random walk. We consider the following problem: when two or more agents meet at a node, they merge into a single agent. As the graph is connected, the agents meet in finite time. We are interested in the time it takes for all agents to merge. More precisely, we study a probabilistic model and we analyse the time complexity of a distributed algorithm for all the agents to merge into a single one.

Keywords: Mobile Agents, Markov Chain, Random Walks

1. INTRODUCTION
With the current trends in distributed algorithms and computing, mobile agents are playing an important role. The stress is laid in using less resources. Acquiring global information of a huge network is a tedious job. The information is scattered around the globe and a mechanism is required to access them easily. A lot of research has been done keeping this in mind, however, there are still a lot of axes to be discovered in this area. In old and traditional distributed systems the data was transferred from node to node which was more difficult due to many network constraints. Mobile agents play an active role i.e., they migrate themselves from one node to another to perform any task.

We are working on models to create and analyse the algorithms using mobile agents. We can define our mobile agent as a light weight entity which does not have any memory. The important aspect of these agents is their displacement property. Owing to the fact that they do not have any memory the displacement method followed by them is random i.e. they decide at any moment \( t \) where to move in the step \( t + 1 \). They are free to make decision on their own and they can also decide to stay at the same place. However, their decision is only limited to a displacement of one step i.e. in their neighboring nodes.

The agents without memory performs the random walk which means that they do not know before hand, where will they go next. Simulating a random walk on a graph requires local information about the graph. That is why random walk has nice global properties. This makes random walk very useful in computations, where limited resources are available to determine global information.

1.1. PROBLEM
We consider multiple agents \( 2 \leq k \leq n \); where \( k \) being the number of agents in the network and \( n \) is the number of nodes. The agents are placed randomly on the nodes. The problem we are studying is that if these agents move in the network then after how much time will all the agents take to meet. We consider the fact that whenever two or more agents meet, they merge and an agent is reduced. We are interested in calculating the time it takes to merge all the \( k \) agents into 1. In this paper, we launch the agents directly in the stationary regime. They displace in the network by doing random walk.

Question: Starting by \( k \) number of agents with \( n \) being the number of nodes of the network, \( k \leq n \), how much time it takes to merge \( k \) agents into 1?

We also observe that the agents will meet with probability 1. As the graph is connected so the agents will merge in finite time.
We are concerned here with a generic problem, which means that we are giving a mechanism for the agents to meet in the network. This problem is of greater interest for a big network where acquiring lot of resources is an issue. So in this case, the agents share or transfer knowledge to other agents hence avoiding the duplicacy in traversal schemes of agents. This means that studying this problem leads to many applications in the field of distributed algorithms.

1.2. AGENTS’ PLACEMENT ON THE NETWORK

Now the question arises as by what mechanism can we place the agents in the network.

1. The agents are placed deterministically on the network; i.e. we place the agents on the nodes according to our requirement or will.
2. The agents are created by the nodes of the network. The nodes choose at random to generate agents based on some timeout mechanism [5].
3. The agents are placed arbitrarily on the nodes hence are completely randomized.
4. The agents are placed arbitrarily on the nodes hence are completely randomized. Moreover, the agents launch their random walk in stationary regime.

The agents are not required to know the placement of the other agents. However, when any two agents come at the same node, they know the existence of the other agent and hence can merge sharing the information etc.

1.3. CONTRIBUTION

We study this problem using the Markov chain technique. The agents perform the random walk and there is a transition in the state of the agents. The transitions are explained in Section 3. We have done analysis considering the probabilistic behaviour of the agents. We did the calculations based on the fact that the mobile agents start their random walk in stationary regime.

1.4. RELATED WORKS

Merging of agents is directly related to rendezvous problem [4]. There are many parameters to be taken into account for studying this kind of problem.

By using the random walk technique, the problem of rendezvous can be solved by introducing the mobile agents. A survey on random walks on graphs can be found in [14]. We are analysing as what happens if we are using multiple agents. Does it improve the efficiency or not? Some works where these agents are used are spanning tree [1], Gathering [11] and Election problem [6].

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 $k$. However, in our problem the mobile agents do not have control, memory and intelligence.

Furthermore, this merging techniques can be used in different applications such as searching for black-holes in the network [8].

1.5. OUTLINE

This paper is organized as follows. Some definitions has been given regarding the stationary regime and the model under study is described in Section 2. The general model for the transitions using Markov chain from 2 to $k$ agents is given in Section 3. Then in Section 4, we study the rendezvous problem under stationary regime. In Section 5, we describe the states of the agents using laws such as Poisson distribution. The agents are placed on the graph and their position is determined by a random variable. The agents are in a unique stationary distribution. In Section 6, we conclude the paper.

2. MODEL & DEFINITIONS

Let $G = (V, E)$ be a connected graph where vertices represent process and edges represent communication links. In the sequel $n$ will denote the size of $G$, i.e. $n = |V|$ and $m = |E|$ be the number of edges in the graph $G$. For any
A mobile agent (MA) is an entity which moves in the graph. The agents have unique identities. As the network is synchronous so all the agents move at one time step $t$. The agent can also decide to stay at the same node at instant $t + 1$. Negligible time is considered in computation.

The agents do not store information about the nodes visited as they do not have memory associated with them. This is useful for applications in which the network is huge. We do not want to increase the network traffic if they require lots of resources. The MAs are not aware of the topology of the network. These agents can follow many models based on various parameters but for our problem the technique we use to analyse their complexity is the model of random walks. When an agent 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. We have kept the hypothesis that the agent can decide to stay at the same node because of the synchronous condition and breaking the symmetry. A simple example of this is that if we take any 2 nodes, and an agent is present on each node then they will never meet. In this paper we use the following definitions and notations from the probability theory (see [9, 10]).

Let $X$ and $Y$ be two random variables. We say that $X$ and $Y$ have the same probability law, and we denote $\mathbb{L}(X) = \mathbb{L}(Y)$ iff $\mathbb{P}(X = x) = \mathbb{P}(Y = x)$, $\forall x$.

Let $k \in \mathbb{N}$, $\lambda \in \mathbb{R}$, and $(X_i)_{1 \leq i \leq n}$ a set of independent r.v. Let $X = (X_1, X_2, \cdot \cdot \cdot, X_n)$, $X$ has a multinomial distribution with parameters $k$ and $(p_i)_{1 \leq i \leq n}$, [9], and we denote it $X \sim \text{mult}(k, (p_i)_{1 \leq i \leq n})$, if $\forall (x_1, x_2, \cdot \cdot \cdot, x_n) \in \mathbb{N^n}$, such that $\sum_{i=1}^{n} x_i = k$,

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \cdot \cdot \cdot, X_n = x_n) = \frac{k!}{x_1!x_2! \cdot \cdot \cdot x_n!} \prod_{i=1}^{n} p_i^{x_i}.$$ 

Let $a$ be any MA in the graph $G$, and $X_a(t)$ the r.v. that represents the $a$’s position in $G$ at time $t \geq 0$. $G$ is a connected graph and for any vertex $v \in V$, $\mathbb{P}(X_a(t+1) = v \mid X_a(t) = v) \neq 0$, then the Markov chain corresponding to the random walk performed by $a$ is irreducible and aperiodic, hence it is ergodic [10], i.e., there is a unique stationary distribution $\pi : V \to [0, 1]$ such that

$$\mathbb{P}(X_a = v) = \pi_v = \frac{d(v) + 1}{2n} = \frac{\rho(v)}{2n}, \forall v \in V.$$

Remark 1 Recall that the word stationary means that if $a$ starts in the stationary distribution, $a$ remains in the stationary distribution.

3. MARKOV CHAIN MODEL

3.1. Configuration Graph

The first formulation of the problem is inspired from [7]. However, their problem is different from ours as they somehow use the centralized control for the agents but for our problem the mobile agents are distributed in the network and there is no central control over the agents. As in [7], a configuration of $k$ agents on the graph $G$ is a multiset of $k$ vertices of $G$. A configuration $\{y_1, y_2, \cdot \cdot \cdot, y_k\}$ describes a placement of $k$ agents on the vertices of $G$, where $y_i$ is the position of agent $a_i$. If $y_i = y_j$, 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 $\nu_0 = \{y_1, y_2, \cdot \cdot \cdot, y_k\}$ with $y_i \neq y_j, \forall i \neq j$ and ends with a terminal configuration $\{y_1, y_2, \cdot \cdot \cdot, y_k\}$ with $y_i = y_j, \forall i \neq j$. If $\nu_p = \{y_1, y_2, \cdot \cdot \cdot, y_k\}$ is the configuration of the process at step $p$ then the next configuration $\nu_{p+1} = \{z_1, z_2, \cdot \cdot \cdot, z_k\}$ of $\nu_p$ satisfies the following conditions:

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

An execution starting at initial configuration $\nu_0$ is a finite sequence of configurations $\{\nu_0, \nu_1, \cdot \cdot \cdot, \nu_l\}$ such that for all $i = 0, 1, \cdot \cdot \cdot, l - 1, \nu_{i+1}$ is a next configuration of $\nu_i$. Configuration $\nu_i$ is called the current configuration at time step $i$. $\nu_l = \{y_1, y_2, \cdot \cdot \cdot, y_k\}$ is a terminal configuration if $y_i = y_j$ for all $i \neq j$, this configuration is noted TERM.
As in [7], 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 

$$E(G_k) = \{(\nu_1, \nu_2) \mid \nu_1 = \{y_1, y_2, \ldots, y_k\} \neq \text{TERM} \text{ and } \nu_2 = \{z_1, z_2, \ldots, z_k\} \text{ is a next configuration of } \nu_1\} \cup \{(\text{TERM, TERM})\}$$

Since $G$ is connected, it is clear that $G_k$ is connected. The behaviour of the set of agents $\{a_1, a_2, \ldots, a_k\}$ 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 $\text{TERM}$, all the other states are transient.

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

In this study, we are interested in 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 $\text{TERM}$ in $G_k$. In terms of random walks, this corresponds to the notion of hitting time $H(\nu_0, \text{TERM})$. (see [9]).

### 3.2. Component 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 behaviour of a random walk on the configuration 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 $\nu_i$ be any configuration, we define the size of $\nu_i$, and we denote it $|\nu_i|$, the number of different vertices in $\nu_i$. We define the component graph $G_k = (V_k, E_k)$ as follows:

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

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 $\nu_i$ in $C$, there is at least a configuration $\nu_2$ 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 $(i, j) \in \{1, 2, \ldots, k\} \times \{1, 2, \ldots, k\}$, we define $p_{i,j}$ as follows:

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

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

$$\Pr(X_{t+1} = C_j \mid X_t = C_i) = p_{i,j}, \forall i, j \in \{1, 2, \ldots, k\} \times \{1, 2, \ldots, k\}$$

We have the following lemma:

**Lemma 2** If $G_k$ is connected then 

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

**Proof:** By induction on $k$. \hfill $\Box$

In [2], the authors show how the study of the particular case of two agents in a graph $G = (V, E)$ allow to obtain an upper bound for the general case of $k \geq 3$ MAs in $G$. In the following study we assume that $G_k$ is connected.

Let $G = (V, E)$ be a given graph of size $n$ and $k \geq 3$ be the number of the MAs in $G$. Let $(X_t)_{t \geq 0}$ be the r.v. which counts the number of the MAs in the graph at time $t$. Clearly, $X_0 = k$ and $\forall t \geq 0$, $X_{t+1} \leq X_t$. The behaviour of the
is an absorbing Markov chain \[12\] with a single absorbing state (the state 0) with the following transition matrix:

\[ P = \begin{pmatrix}
0 & 1 & 2 & \cdots & k-1 & k \\
1 & p_{2,1} & p_{2,2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
k & p_{k,1} & p_{k,2} & \cdots & p_{k,k-1} & p_{k,k}
\end{pmatrix} \]

\( M \) is an absorbing Markov chain \([12]\) with a single absorbing state (the state 0) with the following transition matrix:

\[ P = \begin{pmatrix}
0 & 1 & 2 & \cdots & k-1 & k \\
1 & p_{2,1} & p_{2,2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
k & p_{k,1} & p_{k,2} & \cdots & p_{k,k-1} & p_{k,k}
\end{pmatrix} \]

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

\[ \mathbb{P}(Y_{t+1} = 1 | Y_t = 1) = 1 \]

\[ \mathbb{P}(Y_{t+1} = j | Y_t = i) = \begin{cases} q_{i,j} = p_{i,j} & \text{if } j = i \\ q_{i,j} = \sum_{l=i-1}^{1} p_{i,l} & \text{if } j = i - 1 \\ 0 & \text{otherwise} \end{cases} \]

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

\[ P' = \begin{pmatrix}
1 & 2 & 3 & 4 & \cdots & k-1 & k \\
1 & q_{2,1} & q_{2,2} & 0 & \cdots & 0 & 0 \\
2 & 0 & q_{3,2} & q_{3,3} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
k & 0 & 0 & 0 & \cdots & q_{k,k-1} & q_{k,k}
\end{pmatrix} \]

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

**Lemma 3** For any \( i \in \{1, 2, \ldots, k\} \), if we denote by \( T_i \) (resp. \( T'_i \)) the time for the Markov chain \( M \) (resp. \( M' \)), 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 \( \mathbb{E}(T_i) \leq \mathbb{E}(T'_i), \forall i \geq 2 \).

\[ \blacksquare \]

On the other hand, using the fundamental matrix (see \([12]\)) of the chain \( M' \), it is easy to compute \( \mathbb{E}(T'_i) \) in terms of the transition probabilities \( (q_{i,j}) \) (see \([2]\)) to obtain:

**Lemma 4**

\[ \mathbb{E}(T'_i) = \sum_{i=2}^{k} \frac{1}{q_{i,i-1}} \]

**Remark 5** Previous results can be obtained from \([2]\). The agents perform random walks and the results are obtained on different classes of graphs.
4. THE RENDEZVOUS PROBLEM

4.1. Rendezvous of \( k \) MAs Under the Stationary Regime

In this section, we assume that there are exactly \( k \geq 2 \) MAs in the graph \( G \). We suppose that all the MAs are under the stationary regime, and we study the expected time for all the MAs to merge on a single MA. We denote by \( T_k \) the r.v. which denotes time.

We suppose that \( k \geq 3 \). From the result in Lemma 3 in Section 3, if the component graph \( G_k \) is connected then it suffices to compute the transition probabilities \( q_{i,i-1} \) for all \( i \in \{ 2, 3, \ldots, k \} \). \( q_{i,i-1} \) is the probability that at least two MAs meet somewhere in the graph, that is \( q_{i,i-1} \geq q_{2,1} \) for any \( i \in \{ 2, 3, \ldots, k \} \). Hence

\[
E(T_k) \leq \sum_{i=2}^{k} \frac{1}{q_{2,1}} = \frac{k}{q_{2,1}}.
\]

**Remark 6** In our case, the \( G_k \) is a connected graph. If we do not consider this case then unfortunately our study in previous section will not be valid because of the Markov chain requirements, nevertheless we can upper bound it by using the above mentioned techniques.

Since \( E(T_2) = \frac{1}{q_{2,1}} \), we obtain the following lemma:

**Lemma 7**

\[
E(T_k) = O(kE(T_2)).
\]

4.2. Rendezvous of a Random Number of MAs

In this section, we suppose that the expected number of MAs in the graph is controlled as described in Section 5. That is, the number \( X \) of MAs is a r.v. having a Poisson distribution with parameter \( \lambda = k \). wlog., we assume that \( E(T_0) = E(T_1) = 0 \). Then:

**Theorem 8** If each vertex generates a random number of MAs as described above, and if we denote by \( \tau_k \) the r.v. that represents the time for all the MAs to merge into a single MA, then

\[
E(\tau_k) = O(kE(T_2)).
\]

**Proof:** The proof is based on the results from previous section. Let \( X \) denote the r.v. which counts the total number of MAs in \( G \). Then

\[
E(\tau_k) = \sum_{i \geq 0} E(\tau_k \mid X = i) Pr(X = i) \tag{1}
\]

\[
\leq \sum_{i \geq 0} i \cdot E(T_2) \cdot \frac{k^i}{i!} e^{-k}
\]

Since \( \sum_{i \geq 0} \frac{i^k}{i!} = xe^x \), the theorem follows. \( \square \)

**Rendezvous in cycle-graphs**

Let \( G = (V, E) \) be a cycle of length \( n \geq 3 \) and let \( a \) and \( b \) be two MAs in \( G \). We assume that the two MAs are under stationary regime. For any \( i \geq 0 \), let \( D_t \) denote the distance between the two MAs. It is clear that \( D_t \in \{ 0, 1, 2, \ldots, \lfloor \frac{n}{2} \rfloor \} \) and we can model the behaviour of \( D_t \) by a Markov chain with the following transitions graph:

Then, if we denote by \( E_i \) the expected time for the two MAs to merge into a single MA, conditioned by the fact that the two MAs start at distance \( 0 \leq i \leq \lfloor \frac{n}{2} \rfloor \) one from other then
Figure 1: Possible transitions in a cycle graph (a) at extremeties (b) in general

\[
\begin{align*}
E_0 &= 0 \\
E_1 &= 1 + \frac{1}{9}E_1 + \frac{2}{9}E_2 + \frac{1}{9}E_3 \\
E_2 &= 1 + \frac{1}{9}E_1 + \frac{1}{9}E_2 + \frac{2}{9}E_3 + \frac{1}{9}E_4 \\
E_i &= 1 + \frac{1}{9}E_{i-2} + \frac{2}{9}E_{i-1} + \frac{1}{9}E_i + \frac{2}{9}E_{i+1} + \frac{1}{9}E_{i+2} \quad \forall 2 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1 \\
E_{\lfloor \frac{n}{2} \rfloor - 1} &= 1 + \frac{1}{9}E_{\lfloor \frac{n}{2} \rfloor - 3} + \frac{2}{9}E_{\lfloor \frac{n}{2} \rfloor - 2} + \frac{1}{9}E_{\lfloor \frac{n}{2} \rfloor - 1} + \frac{2}{9}E_{\lfloor \frac{n}{2} \rfloor} \\
E_{\lfloor \frac{n}{2} \rfloor} &= 1 + \frac{1}{9}E_{\lfloor \frac{n}{2} \rfloor - 2} + \frac{4}{9}E_{\lfloor \frac{n}{2} \rfloor - 1} + \frac{1}{9}E_{\lfloor \frac{n}{2} \rfloor}
\end{align*}
\]

After solving the system, we find \(E_i = O(ni)\)

That is, since

\[
\Pr(D_0 = 0) = \Pr(D_0 = \lfloor \frac{n}{2} \rfloor) = \frac{2}{n}
\]

and \(\Pr(D_0 = i) = \frac{1}{n}, \forall 1 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1\)

One can prove:

Lemma 9 If we denote by \(T_2\) the time for the two MAs to merge into a single MA then:

\[
E(T_2) = O(n^2)
\]

**Proof:** The time for the two agents to merge into a single agent is

\[
E(T_2) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} E(T_2 | D_0 = i) \times \Pr(D_0 = i)
\]

Then

\[
E(T_2) = O\left(\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} i\right)
\]

Hence

\[
E(T_2) = O(n^2)
\]
Lemma 10 \textit{The time for the }k\textit{ MAs to merge into a single MA is given by:}

\[ \mathbb{E}(T_k) = O(kn^2) \]

5. \textbf{HOW TO GUIDE LOCALLY THE TOTAL NUMBER OF MAS}

In this section, we show how to control \textit{locally} the \textit{global} number of MAs in \textit{all} the graph. We start by giving some properties of random variables with Poisson distribution.

Let \( X_1, X_2, \ldots, X_n \) be a set of \( n \geq 2 \) independent r.v., such that, \( \forall i \in \{1, 2, \ldots, n\} \), \( X_i \) has a Poisson distribution with parameter \( \lambda_i \). That is:

\[ \Pr(X_i = l) = \frac{\lambda_i^l}{l!} e^{-\lambda_i}, \forall l \geq 0. \] (2)

For any \( (k_1, k_2, \ldots, k_n) \in \mathbb{N}^n \),

\[ \Pr\left((X_1, X_2, \ldots, X_n) = (k_1, k_2, \ldots, k_n)\right) = \prod_{i=1}^{n} \Pr(X_i = k_i) = \prod_{i=1}^{n} \frac{\lambda_i^{k_i}}{k_i!} e^{-\lambda_i} = \prod_{i=1}^{n} \left(\frac{\lambda_i^{k_i}}{k_i!}\right) e^{-\sum_{i=1}^{n} \lambda_i} \quad \text{(3)} \]

On the other hand, let \( X \) be the r.v., \( X = \sum_{i=1}^{n} X_i \), then \( X \) has a Poissonian distribution of parameter \( \lambda = \sum_{i=1}^{n} \lambda_i \).

Once

\[ \Pr\left((X_1, X_2, \ldots, X_n) = (k_1, k_2, \ldots, k_n) \mid X = K\right) = \frac{\Pr\left((X_1, X_2, \ldots, X_n) = (k_1, k_2, \ldots, k_n)\right)}{\Pr(X = K)} \]

and using equation (3), this yields to

\[ \Pr\left((X_1, X_2, \ldots, X_n) = (k_1, k_2, \ldots, k_n) \mid X = K\right) = \frac{K!}{\prod_{i=1}^{n} k_i!} \prod_{i=1}^{n} \frac{\lambda_i^{k_i}}{\lambda^K}, \forall K \geq 0 \quad \text{(5)} \]

Let \( a \) be an MA in \( G \). The Markov chain that describes the random walk of \( a \) is an ergodic one and admits a unique stationary distribution. That is if we denote by \( X_a \) the position of \( a \) in \( G \), under the stationary regime, then

\[ \Pr(X_a = v) = \pi_v = \frac{d(v) + 1}{2m + n}, \forall v \in V. \quad \text{(6)} \]

Assume that we have \( k \) MAs in the graph, and all the MAs are under the stationary regime, then we have the following lemma:

Lemma 11 \textit{Suppose we have }k\textit{ MAs under the stationary regime in the graph, and let } (N_1^k, N_2^k, \ldots, N_n^k) \textit{ be the r.v. defined as follows, } \forall i \in \{1, 2, \ldots, n\}, N_i^k \textit{ is the r.v. which counts the number of MAs in the vertex } v_i. \textit{Then}

\[ (N_1^k, N_2^k, \ldots, N_n^k) \sim \text{mult}\left(k, d(v_i) + 1\right). \]

\textbf{Proof:} For any \( i \in \{1, 2, \ldots, n\}, \forall k_i \in \mathbb{N}, N_i^k = k_i \text{ iff there is exactly } k_i \text{ MAs in the vertex } v_i. \text{ Hence } \forall (k_1, k_2, \ldots, k_n) \in \mathbb{N}^n,

\[ \Pr\left((N_1^k, N_2^k, \ldots, N_n^k) = (k_1, k_2, \ldots, k_n)\right) = \binom{k}{k_1} \binom{k - k_1}{k_2} \ldots \binom{k - \sum_{i=1}^{n-1} k_i}{k_n} \pi_{v_1}^{k_1} \pi_{v_2}^{k_2} \ldots \pi_{v_n}^{k_n} \]
Then, the lemma follows by (6).

We now give the fundamental theorem which will be used in the following section

**Theorem 12**

$$L\left( (X_1, X_2, \cdots, X_n) \mid X = k \right) = L\left( (N_1^k, N_2^k, \cdots, N_n^k) \right).$$

**Proof:** It suffices to take, in (5), \( \lambda_i = \frac{d(v_i)+1}{2m+n} k, \forall i \in \{1, 2, \cdots, n\} \), and then use the result in Lemma 11.

### 5.1. Application

In this paragraph, we show how the results from the above section can be applied to obtain a set of \( k \geq 2 \) MAs already in the stationary regime. We suppose in the sequel that each vertex knows the number \( n \) of vertices, \( m \) the number of edges and \( k \) the number of desired MAs. Consider the following process: each vertex \( v \) generates a random number \( X_v \) of MAs according to the Poisson distribution with parameter

$$\lambda_v = \frac{d(v)+1}{2m+n} k$$

Then we have the following lemma:

**Lemma 13** If each vertex \( v \) in the graph generates a random set of MAs using a Poisson distribution with parameter \( \lambda_v \) as defined above, then all the generated MAs are under the stationary regime.

**Proof:** Direct if we substitute \( \lambda_v \) in (5) and if we apply Theorem (12).

### 6. CONCLUSION AND PERSPECTIVES

In this paper, we have given a general upper bound for a set of \( k \) MAs to merge into a single MA. We also give a technique to control locally the total number of MAs in the graph. Nevertheless, we have supposed that all nodes know \( n, m \) and \( k \). So, the natural question that arises is how can we design a similar technique that does not need all the nodes to know these parameters? We used Poisson distribution, what will happen if we use another distribution? The upper bound obtained in this work for the cycle graph is \( O(kn^2) \). Can we do better in other classes of graphs? A first answer may be obtained if we can exactly compute the transition probabilities of the original Markov chain (in Section 3).

**Bibliography**


