Interacting urns processes : for clustering of large-scale networks of tiny artifacts

Leone, Pierre; Schiller, Elad M.

How to cite

LEONE, Pierre, SCHILLER, Elad M. Interacting urns processes : for clustering of large-scale networks of tiny artifacts. In: Proceedings of the 2008 ACM Symposium on Applied Computing (SAC). Fortaleza (Ceara, Brazil). [s.l.] : ACM Press, 2008. doi: 10.1145/1363686.1364182

This publication URL: https://archive-ouverte.unige.ch//unige:34069

Publication DOI: 10.1145/1363686.1364182

Interacting Urns Processes

for Clustering of Large-Scale Networks of Tiny Artifacts *

[Extended Abstract]

Pierre Leone
Computer Science Department
University of Geneva
pierre.leone@cui.unige.ch

Elad M. Schiller
Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
elad@chalmers.se

ABSTRACT

We analyze a distributed variation on the Pólya urn process in which a network of tiny artifacts manages the individual urns. Neighboring urns interact by repeatedly adding the same colored ball based on previous random choices. We discover that the process rapidly converges to a definitive random ratio between the colors in every urn and that the rate of convergence of the process at a given node depends on the global topology of the network. In particular, the same ratio appears for the case of complete communication graphs. Surprisingly, this effortless random process supports useful applications, such as clustering and pseudo-coordinate computation. We present preliminary numerical studies that validate our theoretical predictions.

Keywords: Pólya Urns, Networks of Tiny Artifacts, Clustering.

1. INTRODUCTION

Designers of distributed algorithms often assume that each node is computationally powerful, capable of storing nontrivial amounts of data and carrying out complex calculations. However, recent technological developments in wireless communications and microprocessors allow us to establish networks consisting of massive amounts of cheap and tiny artifacts that are tightly resource constrained. These networks of tiny artifacts are far more challenging than the traditional networks; on one hand, their relative scale is enormous, while on the other hand, each tiny artifact can only run a millicode that is aided by a miniature memory – possibly as little as a constant number of bits per tiny artifact. Such limitations are not crippling if the system designer has a precise understanding of the tiny artifacts' computational power, and local interaction rules from which global formation emerges.

Urn processes (and more generally processes with reinforcements) is a tool for modeling stochastic processes that show structure emerging. Their aim is to analyze the global structure of a given population given the micro-mechanisms the particular entities are applying. The global structure reflects the ability of the entities to self-organize. For instance, such models can explain the preferential-attachment model in small-world networks, the process of market sharing, interaction of biological entities, etc., based on given micro mechanisms. We refer to [15] and references therein for reinforcement processes, for models based on urn models see [8], and for the emergence of structure we refer to [2].

1.1 Our contribution.

In this paper our aim is to extend the classical urn process to a network of interacting urn processes and to analyze the emerging global formation, which is carried out by a low-level interaction process. The network of tiny artifacts repeats the following operations. Each urn has a multiset of black and white balls, initially one of each color. A given tiny artifact draws a ball from the urn with uniform distribution and returns to the urn the ball along with a new ball of the same color.[1] The tiny artifact interacts with its neighbors on the network. Namely, after returning the two balls, the tiny artifact announces the balls' color to its neighbors, and the neighbors add a new ball of that color to their urns. The selection of the artifact drawing the ball from the urn is random and uniform (any artifact might be selected with the same probability). These assumptions might be implemented by introducing synchronous and centralized coordination of the tiny artifacts or completely distributed mechanism that assumes partial-asynchrony. Indeed, if we assume that before drawing from the urn the tiny artifacts wait for a random time with $\lambda-$exponential distribution, then globally, the tiny artifacts are going to be selected uniformly without any global synchronization. Notice that the algorithm steps are composed of the following operations:

*This work has been partially supported by the Swiss SER Contract N0. C05.0030, and by the ICT Programme of the European Union under contract number FP7-215270 (FRONTS)

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil
Copyright 2008 ACM 978-1-59593-753-7/08/0003 ...$5.00.

[1] This is a classical Pólya urn [9]

(1) an urn ball-drawing operation, (2) a ball announcement operation, and (3) an update of the urn by the drawing tiny artifact and its neighboring urns. Concurrent steps should interleave carefully; we require that at most one artifact among any set of neighbors is accessing its urn at a time, announces its ball drawing, and no neighboring artifact takes a step before all neighboring artifacts update their urns.

This work presents:

- **An analysis** of the interacting urns process, allowing us to demonstrate that the ratio between the numbers of black and white balls in every urn is converging and that these ratios, which are obtained locally, provide global information about the communication graph.
- **Applications** for large-scale networks of tiny artifacts include clustering and virtual coordinate computation. We present preliminary numerical studies that validate our theoretical predictions on the emerging global formations.

The implementation for tiny artifacts requires broadcasting balls on shared communication media. The interacting urn process analysis assumes undisturbed steps, and that all urns are selected with the same probability. We look into different system settings and explain how to implement the interacting urn process. Due to space limits, some parts of the implementation appear in the Appendix.

1.2 Document structure

We start by analyzing the process (Section 2), before presenting the applications (Section 3), and the implementation (Section 4). Our conclusions appear last (Section 5).

2. THE INTERACTING URN PROCESSES

There is no unified way of analyzing the dynamic behavior of the type of systems we consider in this paper. These systems usually cannot be understood with Markovian formalism. Indeed, the dynamics is Markovian but describing the process in this way leads to an intractable set of equations. Relative to these kinds of processes, which are characterized by path-dependence and/or (negative) reinforcement, we can find various ad hoc techniques. Many references are available from the survey [15].

In the following, we analyze the behavior of our interacting urn process by embedding a multitype branching process. This embedding is well known in the case of a single urn [13]. For example, consider balls of different types corresponding to the color of the ball as well as the urn number it belongs to. Balls wait independently for a random exponentially distributed random time and then produce new balls corresponding to its own type and the type of the neighbors of the urn. This corresponds to adding a ball of the same color to the current urn and announcing it to the neighbors. If N is the number of urns composing the network, the state of the multitype branching process is described by the vector

$$Z(t) = \big(Z^1(t), \dots, Z^N(t)\big)^T, \tag{1}$$

with $Z^i(t) = (b^i_t, w^i_t)^T$. The type of a ball is then given by index i referring to the urn number and the color b, w. To simplify the notation, we start by writing down the equations for the total population belonging to the urns $Z^i(t) = b^i_t + w^i_t$. If we consider that the particles are evolving independently, the lifetime of the particles composing the same urn has to be inversely proportional to the total number of balls in the urn in order to ensure that particles split/die at a constant rate to mimic the interacting urn process in which a particular urn is selected with uniform distribution. Hence, the lifetime of particles in urn i is exponentially distributed with parameter

$$\lambda^i(t) = \frac{\lambda}{b^i_t + w^i_t} \approx \frac{1}{t(\deg(i)+1)}. \tag{2}$$

Since it is clear that $b^i_t + w^i_t \approx \lambda(\deg(i)+1)t$ as t is large. With $\mathbf{e_i} \in \Re^N$ the vector with all entries at zero except the ith, the generating function

$$F^i(t,\mathbf{s}) = E(s_1^{Z^1(t)} \dots s_N^{Z^N(t)} \mid Z(t) = \mathbf{e_i}), \tag{3}$$

can be shown to satisfy the forward Kolmogorov equation

$$\frac{d}{dt}F^i\big(t,\mathbf{s}\big) = \sum_{k=1}^{N} \lambda^k(t)\big(\mathbf{s}^{\mathbf{n_k}+\mathbf{e_k}} - s_k\big)\frac{\partial}{\partial s_k}F^i\big(t,\mathbf{s}\big), \tag{4}$$

where the jth entry of the vector $\mathbf{n_k}$ is 0 or 1 to indicate that the jth node is a neighbor or not of k. This vector describes the topology of the communication graph. Equation (4) can be found in a more general setting in [6], chapter V. The dependence with time of the parameters $\lambda^k(t)$, see (2), expresses the interdependence between the urn processes. In our situation, we can use the asymptotic limit approximation (2). Indeed, if we proceed to an exponential time change, the time dependence is suppressed. Writing $F^i(\log(t),\mathbf{s}) = \tilde{F}^i(t,\mathbf{s})$ and using the asymptotic limit (2), we get

$$\frac{d}{dt}F^i\big(t,\mathbf{s}\big) = \sum_{k=1}^{N} \frac{1}{(\deg(i)+1)}\big(\mathbf{s}^{\mathbf{n_k}+\mathbf{e_k}} - s_k\big)\frac{\partial}{\partial s_k}F^i\big(t,\mathbf{s}\big). \tag{5}$$

The forward Kolmogorov equation (5) is one of a multitype branching process with constant rate ($\lambda^k(t) = \lambda^k$). From [3, 13] we know that

$$\lim_{t\to\infty} Z(t,\omega)e^{-\lambda_1 t} = W(\omega)u \tag{6}$$

where $Z(t,\omega)$ is the state vector of our interacting urn process (1) in which we make explicit the underlying probability space by writing the sample point ω; $W(\omega)$ is a random variable and u is the left eigenvector of the matrix A corresponding to the largest eigenvalue λ_1. Hence, because of the exponential time change, the solution of (4) is going to converge like

$$\lim_{t\to\infty} \frac{Z(t,\omega)}{t^{-\lambda_1}} = W(\omega)u. \tag{7}$$

Remark : The use of the asymptotic limit (2) to derive equation (5), which is known to converge like (6), and then deduce that the solution to (4) converges to (7) is subject to caution. However, it is motivated by the fact that the convergence to (5) of (2) is independent of the initial condition.[2] Actually, the result we obtain in this section can be formally derived with the techniques introduced in [10].

From (5) we know that given an initial urn content vector $(Z^1(0), \dots, Z^N(0))^T$, the expected population is given by

$$(Z^1(0), \dots, Z^N(0))M(t) = (Z^1(0), \dots, Z^N(0))\exp(At),$$

with the matrix A given by

$$a_{ij} = \frac{\delta_{ij}}{\deg(i)+1}, \quad i,j = 1, \dots, N. \tag{8}$$

[2] Except perhaps of the law of $W(\omega)$ that we do not consider.

The convergence rate of the process is determined by the second largest eigenvalue.

PROPOSITION 1. *The eigenvalues $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n$ of the matrix A are given by $\lambda_1 = 1 - \mu_i$ with $0 = \mu_1 < \mu_2 \leq \ldots \leq \mu_n$ the eigenvalues of the Laplacian of the communication graph of the network.*

PROOF. Let Ad be the adjacency matrix of the communication graph ($a_{ij} = 1_{i\sim j}$) and D the diagonal matrix with $d_{ii} = \deg(i)$. The matrix A is given by $A = (I+D)^{-1}(I+A_d)$. Algebraic manipulations show that if x, λ are right eigenvector and eigenvalue of A, then $x, 1-\lambda$ are right eigenvector and eigenvalue of the Laplacian $D - A_d$ of the communication graph. The result follows since we assume that the communication graph is connected (then the eigenvalue 0 is simple) and by known properties of the spectrum of the Laplacian matrix. □

This proposition shows that the dominant eigenvalue of the matrix A is 1 and one can check that the corresponding left eigenvector is given by $(\deg(1)+1, \ldots, \deg(N)+1)^T$.

PROPOSITION 2. *Let x be the left eigenvector of A corresponding to the eigenvalue 1, then $u = (D+I)^{-1}x$ is a right generalized eigenvector of (A_d, D), i.e., it satisfies $A_d u = Du$.*

PROOF. The proof uses the same decomposition of the matrix A as in the previous proposition and proceeds by direct computations using the fact that the matrices $D+I$ and A_d are symmetric. □

Generalized eigenvectors are useful for spectral graph drawing. Actually, as discussed in [11], generalized eigenvectors corresponding to the generalized eigenvalues smaller than the dominant one provide coordinates for drawing the graph in the plane. Numerical evidence supports the fact that such generalized eigenvectors provide better coordinates than the eigenvectors of the Laplacian matrix. This proposition will support our application of the interacting urn process to clustering.

We now consider what does happen if we distinguish black and white balls in a same urn. Actually, the analysis previously presented carries on, the rate of split/die of the particles living in a given urn of different colors being asymptotically independent since $b_t^i + w_t^i \to t(\deg(i)+1)t$, see (5).

THEOREM 1. *The scaled interacting urn process populations converge to a random vector which is proportional to the left eigenvector of the matrix A corresponding to the maximal eigenvalue* 1*, see* (7)*. The ratio of the black balls among the total population of a given urn, denoted X_t^i, converges to a mean value*

$$X_\infty^i = \frac{1}{deg(i)+1} \sum_{j=1}^{N} \delta_{ij} X_\infty^j, \qquad (9)$$

which is equivalent to

$$X_\infty^i = \frac{1}{deg(i)} \sum_{j=1}^{N} 1_{i\sim j} X_\infty^j.$$

PROOF. The equivalence between the two expressions above follows from simple computation; we point out that the difference between the two sums is that in the first one we take into account the term X_∞^i while we do not in the second. The populations of black balls in the urn b_∞^i satisfies; see (7)

$$b_\infty^i = \sum_{j=1}^{n} \frac{\delta_{ij}}{\deg(j)+1} b_\infty^j.$$

Moreover, the total populations composing the urns satisfy

$$\frac{b_\infty^i + w_\infty^i}{\deg(i)+1} = \frac{b_\infty^j + w_\infty^j}{\deg(j)+1},$$

because of the asymptotic limit (2). Hence,

$$X_\infty^i = \frac{b_\infty^i}{b_\infty^i + w_\infty^i} = \sum_j \frac{\delta_{ij} b_\infty^j}{(\deg(j)+1)(b_\infty^i + w_\infty^i)}$$

$$= \sum_j \frac{\delta_{ij} b_\infty^j}{(\deg(i)+1)(b_\infty^j + w_\infty^j)} = \frac{1}{\deg(i)+1} \sum_j \delta_{ij} X_\infty^j.$$

□

Broadly speaking, it is hard to get general results on the convergence rate of the interacting urn process because it is related to random processes (e.g., processes with reinforcement [15]). The process convergence rate depends on the value of the second largest eigenvalue of A. The difficulty of asserting general results is mainly because such results depend on the topology of the interactions, i.e., the matrix A. However, the case where the topology of the graph of interactions is a complete graph is easy to understand, because the completeness implies identical urn content and the process is similar to the classical pólya urn. This can motivate us to investigate the applications of the interacting urn process in clustered networks, since nodes that share many links are apt to develop similar urn content. Another *folk* result concerning processes similar to the interacting urn process, is that the time for convergence to the definite value is usually very large and not numerically observable. However, some significant values may emerge quickly from the dynamic process. Moreover, notice that due to the probabilistic nature of the interacting urn process, fault tolerances is implied; there is an inherent recovery after the loss of a ball, say, due to communication interferences.

3. APPLICATIONS

We now turn to describe two of the many possible applications of the interacting urn process in networks of tiny artifacts.

3.1 Cluster formation

Spectral graph drawing considers the entries of the generalized eigenvector of (A_d, D) corresponding to the second largest eigenvalue as $1d$ coordinate of the node as an efficient heuristic, see [11].[3] It is then natural to cluster nodes that are close in the $1d$ drawing of the graph. We have shown that the population vector $(b_t^1, \ldots, b_t^N)/t$ converges, with the left eigenvector of the matrix A corresponding to the dominant eigenvalue 1, see equation 7, and that $b_t^i + w_t^i \to t(\deg(i)+1)$. The ratio $X_t^i = b_t^i/(b_t^i + w_t^i)$ is then converging to the generalized eigenvector of (A_d, D) corresponding to the dominant eigenvalue 1 by proposition 2, with a rate of convergence depending on the second largest eigenvalue. However,

[3] By considering successive generalized eigenvectors of (A_d, D) one can also obtain multidimensional representation of the graph.

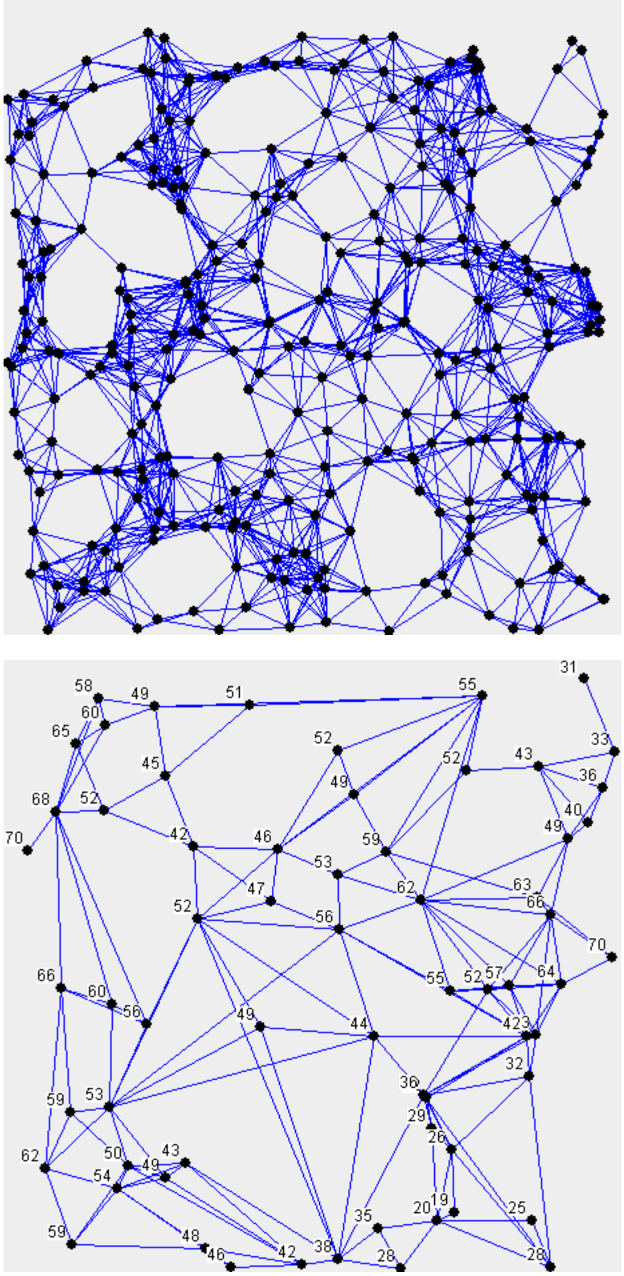

Figure 1: Clusters obtained with the interacting urn process. The top depicts the communication graph ($n = 300$ and $r = 0.12$). The bottom depicts the clusters with their assigned numerical values. For each cluster, we choose a representative node and connect representative nodes of other clusters, whenever there are neighboring nodes belonging to both clusters.

by direct computation, one can check that this generalized eigenvector is $(1, \dots, 1)^T$, and by considering the difference $X_t^i - X_t^j$ we get an approximation of the component of the generalized eigenvalue of (A_d, D) corresponding to the second largest eigenvalue, which is related to the distance of the nodes in the $1d$ drawing of the graph. The preceding analysis suggests running the interacting urn process and clustering neighbor nodes whose difference $X_t^i - X_t^j$ is below a threshold.

A preliminary numerical study that validates our theoretical prediction is presented in Figure 1. In the experiment, the schedule of nodes interaction is generated uniformly at random. The time to stop the process was obtained while the clustered network stopped evolving visually, after about 20 draws per node; however, the global formation starts to emerge after 15 draws per node. In Figure 1, we observe that the algorithm behaves as expected, and a global cluster formation emerges.

3.2 Pseudo Coordinates

In very large-scale networks, it is not possible to register the location of all nodes manually, or to equip all tiny artifacts with GPS units [7]. Nevertheless, position-awareness is of great importance for many applications, e.g., geo-routing (see [12]). We assume that merely a few *landmark nodes* (so-called anchor) have a registered position by some means, e.g., a GPS unit. We aim at letting all others nodes derive a coordinate system through connectivity information. Our approach is similar to that of Wattenhofer et al. [17], however, we are interested in using the interacting process as a heuristic. Consider the following process, in which the landmark nodes keep their coordinates unchanged during the entire process, and other nodes choose an initial position randomly. Then, repeatedly, the nodes transmit their positions, collect the coordinates of their neighbors and assign themselves to the barycentric coordinates. The processes converge, producing pseudo-coordinates for the nodes.

We wish to emulate a similar process to the one described above, and let the interacting urn process estimate the mean of the barycentric neighboring values. Therefore, nodes maintain two urns, one for the x-axis and one for the y-axis, each urn composed of black and white balls. The position (x, y) of a landmark node corresponds to the color ratio of balls to be sent to their neighbors, i.e., the urn with a constant number of x white balls and y black balls. Other nodes start with one black ball and one white ball in their urn. We let the interacting urn process run before using the content of both urns for producing the coordinates by taking the integer part of a factor of the color ratio in every urn. We identify nodes with the same integer coordinates as belonging to the same cluster, because of their similarity to the clustering application above.

The obtained coordinates are useful for geo-routing, as clearly presented in Figure 2. Similar to the clustering application, the time to stop the process was obtained while the pseudo-coordinate stopped evolving visually, after about 20 draws per node; however, the global formation starts to emerge after 15 drawings per node. Lastly, we note that the virtual coordinates (see [12]) are mainly consistent with the real positions. However, this is not expected in our experiment because the topology of the communication graph plays an important role in obtaining the virtual coordinates, whereas our aim is for any coordinates that merely facilitate geo-routing.

4. THE IMPLEMENTATION

Our analysis of interacting urn processes assumes that time is continuous. However, in practice, clock mechanisms

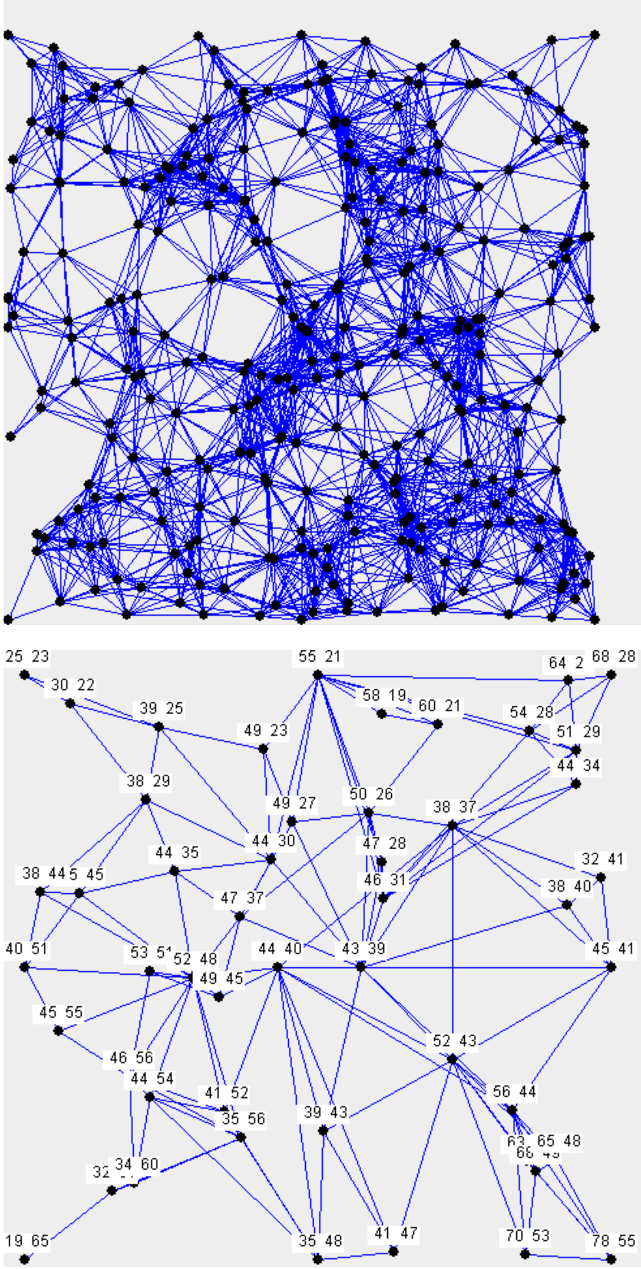

Figure 2: Pseudo-coordinates obtained with two interacting urn processes. On the top we see the communication graph ($n = 300$ and $r = 0.12$) with 8 landmark nodes at $(0.0, 0.0)$, $(0.0, 1.0)$, $(0.5, 0.0)$, $(0.5, 0.5)$, $(0.5, 1.0)$, $(1.0, 0.0)$, $(1.0, 0.5)$, $(1.0, 1.0)$.The bottom depicts the clusters with their pseudo-coordinates. The numbers are the colors' ratio multiplied by 100. See Figure 1 for the clusters' description.

are discrete. Fortunately, when considering a stochastic process evolving in continuous time, it is always possible to conduct a discrete process by considering merely the produced successive events and ignore any reference to continuous time. The transformation from continuous to discrete

```
Types
  colors: { balck, white }

Variables
  urn: multiset of colors, initiialy ⟨balck, white⟩

External functions
  fetch() / deliver() / transfer(): data link layer interface
  get_lock() / release_lock(): lock interface
  set-timer() / timer_expiried(): timmer interface
  select(): a uniform selection

Macro
  time2wait(): choose time to wait using $\lambda-$exponential distribution

Upon deliver(⟨ball⟩)
  urn ← urn ∪ { ball }

Upon timer_expiried()
  let fail = get_lock()
  if not fail then
    let ball = select(urn)
    transfer(m)(⟨ ball ⟩)
    urn ← urn ∪ { ball }
  release_lock()
  set-timer(time2wait())

Constants
  $D$ = upper bound on neighboring nodes' number
```

Figure 3: The interacting urn algorithm.

stochastic model is known as *discrete skeleton* (see [5]).

Another assumption that we make in our analysis is about the instant algorithm's step (i.e., it takes no time to draw a ball, announce it, and let the drawing tiny artifact and the neighboring tiny artifacts update their urns). In real distributed systems, this assumption does not hold and the concurrent algorithm's steps can be non-serializable. Therefore, we require that at any time, there is at most one tiny artifact that takes the algorithm's step.

We present the pseudocode for a possible implementation in Figure 3. The code makes use of $\lambda-$exponential distributed when waiting a random timeout period as well as a distributed locking mechanism for assuring the above requirement. Chockler et al. [4] explain how to implement a mechanism that enables implementation of a distributed locking mechanism in synchronous settings with different kinds of collision detectors. Due to the space limits of this paper, we present in the Appendix a self-stabilizing mechanism for implementing a distributed locking mechanism in partially synchronous system settings using perfect and imperfect collision detectors. We believe that it might be possible to extend the work of Panconesi et al. [14] to allow a lightweight mechanism for repeated test-and-set in asynchronous settings.

5. DISCUSSION

5.1 Related work

Urn processes (and more generally processes with reinforcements) is a tool for modeling stochastic processes that show emerging structures. Their aim is to analyze the global structure of a given population given the micro-mechanisms the particular entities are applying. The global structure reflects the ability of the entities to self-organize. Such models can explain the preferential-attachment model in small-world networks, the process of market sharing, interaction of biological entities, etc., based on given micro mechanisms. We refer to [15] and references therein for reinforcement processes, for models based on urn models see [8], and for the emergence of structure we refer to [2].

Tishby and Slonim [16] use random processes for network clustering. The authors use a Markov process and the clusters are built by considering the decay of mutual information. The information changes with time and during an appropriate period, the mutual information is relevant for clustering. After this period, the mixing property of the Markov process destroys the emerged structure. The random processes we suggest converge so slowly that the mixing property is useful for tolerating various faults. Nevertheless, the process obtains the significant values rapidly.

Angluin et al. [1] define *urn automata* that include a state controller, and an urn containing balls with a finite set of colors. Tiny artifacts can implement the urn automata with provable guarantees regarding their computational power and, by that, allowing the exact analyses of the interaction process among artifacts. This work aims at understanding the possibilities of the urn automata to emerging global formations using miniature algorithms and diminutive resources.

5.2 Conclusions

We are interested in simplifying the design of tiny artifacts, and bridging the gap between these future networks and existing ones. Existing implementations, say, for sensor networks, often use protocols that assume traditional system settings that require resources that tiny artifacts do not have. Alternatively, when the designers do not assume traditional system settings, they turn to improving performance and reducing resource consumption by using probabilistic algorithms. However, designers that do not consider implementation explicitly do not specify the exact computational power required for each node. In some cases, the implementation requires storing non-trivial quantities of data.

This paper shows the existence of an infrastructure for tiny artifacts that requires merely $O(\log D)$ of space, transmits 1-bit at a time, and still rapidly facilitates reasonable clustering with efficient geo-routing.

5.3 Acknowledgments

This work would not have been possible without the contribution of Paul G. Spirakis in many helpful discussions, ideas, and analysis. Many thanks to Edna Oxman for improving the presentation.

6. REFERENCES

[1] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Urn automata. Technical Report YALEU/DCS/TR1280, Yale University Dep. of Computer Science, P.O. Box 208205, New Haven, CT 06520–8285, USA, November 2003.

[2] B. Arthur, Y. Ermoliev, and Y. Kaniovski. Path dependent processes and the emergence of macrostructure. In B. Arthur, editor, *Increasing returns and path dependence in the economy*, chapter 3, pages 33–48. The University of Michigan Press, Ann Arbor, MI, 1994.

[3] K.B. Athreya. Some Results on Multitype Continuous Time Markov Branching Processes. *The Annals of Mathematical Statistics*, 39(2):347–357, 1968.

[4] Gregory Chockler, Murat Demirbas, Seth Gilbert, Nancy A. Lynch, Calvin C. Newport, and Tina Nolte. Reconciling the theory and practice of (un)reliable wireless broadcast. In *ICDCS Workshops*, pages 42–48. IEEE Computer Society, 2005.

[5] P. Guttorp. *Stochastic Modeling of Scientific Data.* Chapman & Hall/CRC, 1995.

[6] T. Harris. *The Theory of Branching Processes.* Springer-Verlag, New York, 1963.

[7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS Theory and Practice.* New York, NY, 2001.

[8] N. L. Johnson and S. Kotz. *Urn Models and Their Applications: An Approach to Modern Discrete Probability Theory.* Wiley, New York, 1977.

[9] Norman L. Johnson and Samuel Kotz. *Urn Models and Their Applications: An Approach to Modern Discrete Probability Theory.* Wiley, New York, 1977.

[10] FC Klebaner. Asymptotic Behaviour of Markov Population Processes with Asymptotically Linear Rate of Change. *Journal of Applied Probability*, 31(3):614–625, 1994.

[11] Yehuda Koren. On spectral graph drawing. In Tandy Warnow and Binhai Zhu, editors, *COCOON*, volume 2697 of *Lecture Notes in Computer Science*, pages 496–508. Springer, 2003.

[12] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer. Virtual coordinates for ad hoc and sensor networks. *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 8–16, 2004.

[13] P.E. Ney and K.B. Athreya. *Branching Processes.* Courier Dover Publications, 2004.

[14] Alessandro Panconesi, Marina Papatriantafilou, Philippas Tsigas, and Paul Vitányi. Randomized naming using wait-free shared variables. *Distrib. Comput.*, 11(3):113–124, 1998.

[15] Robin Pemantle. A survey of random processes with reinforcement. *PROBABILITY SURVEYS*, 4:1, 2007.

[16] Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, pages 640–646. MIT Press, 2000.

[17] Mirjam Wattenhofer, Roger Wattenhofer, and Peter Widmayer. Geometric routing without geometry. In Andrzej Pelc and Michel Raynal, editors, *SIROCCO*, volume 3499 of *Lecture Notes in Computer Science*, pages 307–322. Springer, 2005.