The Dynamics of Adaptive Networked Societies of Tiny Artefacts *

Ioannis Chatzigiannakis Paul G. Spirakis

Computer Technology Institute (CTI) and CEID, University of Patras, Greece
E-mail: {ichatz,spirakis}@cti.gr

Abstract: In the near future, it is reasonable to expect that new types of systems will appear, of massive scale that will operating in a constantly changing networked environment. We expect that most such systems will have the form of a large society of tiny networked artefacts. Angluin et al. [1] introduced the notion of "Probabilistic Population Protocols" (PPP) in order to model the behavior of such systems where extremely limited agents are represented as finite state machines that interact in pairs under the control of an adversary scheduler. We propose to study the dynamics of Probabilistic Population Protocols, via the differential equations approach. We provide a very general model that allows to examine the continuous dynamics of population protocols and we show that it includes the model of [1], under certain conditions, with respect to the continuous dynamics of the two models. Our main proposal here is to exploit the powerful tools of continuous nonlinear dynamics in order to examine the behavior of such systems. We also provide a sufficient condition for stability.

1 Introduction

In the near future, it is reasonable to expect that new types of systems will appear, designed or emerged, of massive scale, expansive and permeating their environment, of very heterogeneous nature, and operating in a constantly changing networked environment. Such systems are expected to operate even beyond the complete understanding and control of their designers, developers, and users. Although they will be perpetually adapting to a constantly changing environment, they will have to meet their clearly-defined objectives and provide guarantees about certain aspects of their own behavior.

We expect that most such systems will have the form of a very large society of networked artefacts. Each such artefact will be unimpressive: small, with limited sensing, signal processing, and communication capabilities, and usually of limited energy. Yet by cooperation, they will be organized in large societies to accomplish tasks that are difficult or beyond the capabilities of todays conventional centralized systems. These systems or societies should have particular ways to achieve an appropriate level of organization and integration. This organization should be achieved seamlessly and with appropriate levels of flexibility, in order to be able to achieve their global goals and objectives.

Our envisioned systems have an identified purpose (which depends on the application). Adaptation should continue to serve this purpose. This means that sudden variations of external service requests or enviromental physical conditions or of motion of network nodes should not stop the system from serving its goal. Instead the system must continue to operate in a set of desired states with maintained, or gracefully degraded or even improved quality of service.

The scale and nature of these systems requires naturally that they are pervasive. This inherent characteristic is both a benefit and a constraint for the study of such systems. The ability of networked societies of small artefacts to adapt is composed of two almost orthogonal dimensions, each with its own issues and objectives:

- The ability for internal continual self-organizational of the network.
- The ability to adapt to environmental changes in a dynamic way. In particular, for systems deployed to achieve particular goals, this adaptability should also address the needs, constraints, and commands of its users.

1.1 The need for a foundational approach

We believe that it is important to establish the foundations of adaptive networked societies of small or tiny heterogeneous artefacts. We need to develop an understanding of such societies that will enable us to establish their fundamental properties and laws, as well as their inherent trade-offs. Our foundational approach to adaptation includes effort to devise schemes to measure the quality of adaptation and the degree of optimality of adaptation. Some measures are (a) how fast the network adapts to the environmental changes (response time) and (b) how much the system pays in terms of energy and communication overhead for a quick adaptation. Less obvious but important measures include:

*This work has been partially supported by the IST Programme of the European Union under contract number ICT-2008-215270 (FRONTS).

(a) how much to adapt (and the limits of adaptability), (b) how much the system pays in overhead in order *to stay prepared for possible future adaptations* (cost of maintaining global structures, economy of enegy, cost of continuous readyness and awareness).

Establishing a science of adaptive organizations of large nets of small or tiny artefacts will allow researchers to use models, laws and fundamental properties in order to further investigate the operation of such systems in particular cases, specific to real application scenarios. System designers and solution developers will be able to apply the fundamental principles during the design process in order to investigate the performance of the systems under development and better understand the inherent trade-offs of the resulting network.

In terms of the ability for internal continual self-organizational of the network, it is necessary to provide mechanisms that will deal with adversarial situations; starting from the physical layer, continuing with the (MAC) data link, and network layer all the way to the application layer. Such mechanisms will be tailored to small devices implementations of proxy and scaled-down solutions. They can be used directly to ensure the adaptiveness, self-stabilization and self-organization design criteria or they can be integrated as part of larger systems in order to deal with specific problems. So, we forsee a future contribution to embedded systems design.

Regarding the ability to adapt to environmental changes in a dynamic way, system developers of Future Open Systems (e.g., robotic systems) will be able to better understand the ability of systems to achieve particular goals, given the actual conditions of the environment. They will be able to classify the resulting systems and better understand the degree of possible control over the environment and the actual constraints. It is thus important to provide algorithms that can be used in systems of autonomous (or semi-autonomous) mobile objects for a variety of activities dealing with robotic exploration, including communication, exploration, scheduling robots to jobs, 3-D visualization. Such algorithms for maintaining connectivity, exploration strategies for many robots, local scheduling can be directly used in future system or integrated in order to deal with the vital challenges of the dynamic aspects of pervasive networks.

1.2 Research challenges

The internal self-organization requires to address at least two problems: (a) how to continually adapt the communication infrastructure and (b) how to achieve "self-stability", which allows effective recovery from transient unexpected faults. We believe that the second problem is of central importance because self-stabilization is an indispensable property of the systems under examination.

The adaptation to the environment and to the needs of users requires to address the following problems: (a) how to achieve distributed cooperation, (b) how the system "tribes" discover and track resources, (c) how the net reacts to imposed, uncontrolled dynamicity (such as externally imposed movements of the artefacts because, e.g., they follow ocean currents or are attached to humans), and (d) the extremely important objective of how trust develops or emerges in the whole net or its parts.

Both kinds of adaptational ability require to be able to cope, on one hand, with all kinds of threats, faults, and attacks, and on the other hand, to be able to establish and maintain trust to the humans and to the other parts of the net. Adaptive security and trust in dynamic settings are tasks we need to address in both lines of objectives of our research.

2 Population Protocols

Angluin et al. [1, 2] introduced the notion of a computation by a population protocol to model such distributed systems in which individual agents are extremely limited and can be represented as finite state machines. In their model, finite-state, and complex behavior of the system as a whole emerges from the rules governing pairwise interaction of the agents. The computation is carried out by a collection of agents, each of which receives a piece of the input. These agents move around and information can be exchanged between two agents whenever they come into contact with each other. The goal is to ensure that every agent can eventually output the value that is to be computed (assuming a fairness condition on the sequence of interactions that occur).

The network is modeled as a complete graph G where vertices represent nodes and edges represent communication links between nodes. We use the letter n to denote $|V|$, the number of nodes in the network. Each node is capable of executing a "process" which consists of the following components:

- Q, a finite set of states.
- X, a nonempty subset of Q, known as the inital states or start states.
- T, a transition function $(\alpha, \beta) \mapsto (\alpha', \beta')$ that maps ordered pairs of states to ordered pairs of states.
- M, a message-generation function mapping states to elements of some fixed message alphabet Y, known as output messages.

We assume that the nodes of the network are numbered from 1 to n. The processes associated with the nodes of G do not know their indices.

A state assignment of a system is defined to be an assignment of a state to each process in the system. An execution of the system is defined to be a sequence $C_0, C_1, C_2, \ldots$ of configurations (where C_0 is the initial configuration) such that for each i, $C_i \mapsto C_{i+1}$. We say that C_i can reach C_{i+1}, if there exists distinct processes u and v such that the state of u in C_i, $C_i(u) = \alpha$, and the state of v in C_i, $C_i(v) = \beta$, the transition function T specifies $(\alpha, \beta) \mapsto (\alpha', \beta')$ where $C_{i+1}(u) = \alpha'$ and $C_{i+1}(v) = \beta'$ and $C_{i+1}(k) = C_i(k)$ for all k other than u and v. In this interaction we say that u is the initiator and v is the responder.

An execution converges to an output $y \in Y$ if there exists an i such that for every $j \geq i$, the message generation function applied to every state occuring in C_j is y. The process do not know when convergence to a common output has reached, and protocols are generally designed not to halt.

An execution is fair if for any C_i and C_j, such that $C_i \mapsto C_j$ and C_i occurs infinitely often in the execution, C_j also occurs infinitely often in the execution.

2.1 Probabilistic Population Protocols

In [1] they also proposed a natural probabilistic variation of the standard population protocol model, in which finite-state agents interact in pairs under the control of an adversary scheduler. In this variant, interactions that occurs between pairs of agents are chosen uniformly at random. We call the protocols of [1] by the term "Probabilistic Population Protocols" (PPP).

A probabilistic protocol is defined by a particular probability distribution over executions from a given configuration where C_{i+1} is generated from C_i by drawing an order pair (u, v) of processes independently and uniformly, applying the transition function T to $(C_i(u), C_i(v))$ and updating states of u and v accordingly to obtain C_{i+1}. The execution obtained this way will be fair with probability 1.

A protocol stably computes a predicate P on multisets of elements of X if for any input configuration C, every fair execution of the protocol starting with C converges to 1 if P is true on the multiset of inputs represented by C, and converges to 0 otherwise.

In [3] they presented fast algorithms for performing computations in this variation and showed how to use the notion of a leader in order to efficiently compute semilinear predicates and in order to simulate efficiently LOGSPACE Turing Machines. [4] studied the acquisition and propagation of knowledge in the probabilistic model of random interactions between all paris in a population (conjugating automata).

3 Switching Probabilistic Protocols

We provide a very general model that allows to examine the continuous dynamics of population protocols and we show that it includes the model of [1], under certain conditions, with respect to the continuous dynamics of the two models. We characterize the dynamics of population protocols by examining the rate of growth of the states of the agents as the protocol evolves. We imagine here a continuoum of agents. By the law of large numbers, one can model the undelrying aggregate stochastic process as a deterministic flow system. Our main proposal here is to exploit the powerful tools of continuous nonlinear dynamics in order to examine questions (such as stability) of such protocols.

The network is modeled as a complete graph G where vertices represent nodes and edges represent communication links between nodes. We use the letter n to denote $|V|$, the number of nodes in the network. Each node is capable of executing an "agent" (or process) which consists of the following components:

- K, a finite set of states. We use the letter k to denote $|K|$.
- X, a nonempty subset of K, known as the inital states or start states.

We consider a large population of n agents. Let $q \in K$ be a state of the agent and let n_q the number of agents that are on the given state p. Then the total population size is $n = \sum_{i=1}^{k} n_i$. The proportion of agents that are at state q is $x_q = \frac{n_q}{n}$. We call x_q the *density* of q. In the sequel $q = q_i$, where $i \in \{1, 2, \ldots, k\}$.

A state assignment of a system is defined to be an assignment of a state to each agent in the system. A *configuration* C is a map from the population to states, giving the current state of every agent. The population state then, at time t, can be described via a vector $\vec{x}(t) = (x_1(t), \ldots, x_k(t))$. Here $x_i(t) = \frac{n_i}{n}$, $i = 1 \ldots k$.

In the sequel we assume that $n \to \infty$. We are interested, thus, in the evolution of $\vec{x}(t)$ as time goes on. We use a different model (compared to [1]) for describing a protocol P. We imagine that all agents in the population are infinitely lived and that they interact forever. Each agent sticks to some state in K for some time interval, and now and then *reviews* her state. This depends on $\vec{x}(t)$ and may result to a change of state of the agent. Based on this concept, a *switching population protocol* consists of the following two basic elements (specifications):

1. A specification of the *time rate* at which agents in the population review their state. This rate may depend on the current, "local", performance of the agent's state and also on the configuration $\vec{x}(t)$.

2. A specification of the *switching probabilities* of a reviewing agent. The probability that an agent, currently in state q_i at a review time, will *switch* to state q_j is in general a function $p_{ij}(\vec{x}(t))$, where $p_i(\vec{x}) = (p_{i1}(\vec{x}), \ldots, p_{ik}(\vec{x}))$ is the resulting distribution over the set K of states in the protocol.

In a large, finite, population n, we assume that the review times of an agent are the "birth times" of a Poisson process of rate $\lambda_i(\vec{x})$. At each such time, the agent i selects a new state according to $p_i(\vec{x})$. We assume that all such Poisson processes are independent. Then, the aggregate of review times in the sub-population of agents in state q_i is itself a Poisson process of birth rate $x_i\lambda_i(\vec{x})$. As in the probabilistic model of [1] we assume that state switches are independent random variables accross agents. Then, the rate of the (aggregate) Poisson process of switches from state q_i to state q_j in the whole population is just $x_i(t)\lambda_i(\vec{x}(t))\, p_{ij}(\vec{x}(t))$.

When $n \to \infty$, we can model the aggregate stochastic processes as deterministic flows (see, e.g., [7, 8]). The outflow from state q_i is $\sum_{j \neq i} x_j \lambda_j(\vec{x})\, p_{ij}(\vec{x})$. Then, the rate of change of $x_i(t)$ (i.e. $\frac{dx_i(t)}{dt}$ or $\dot{x}_i(t)$) is just

$$\dot{x_i} = \sum_{j \in K} x_j p_{ji}(\vec{x})\, \lambda_j(\vec{x}) - \lambda_i(\vec{x})\, x_i \qquad (1)$$

for $i = 1, \ldots, k$.

We assume here that both $\lambda_i(\vec{x})$ and $p_{ij}(\vec{x})$ are Lipschitz continuous functions in an open domain Σ containing the simplex Δ where

$$\Delta = \left\{ (x_i, \ldots, x_k) : \sum_{i=1}^{K} x_i = 1\,, \qquad x_i \geq 0\,,\ \forall i \right\}$$

By the theorem of Picard-Linderlöf (see, e.g., [6] for a proof), Eq. 1 has a *unique* solution for any initial state $\vec{x}(0)$ in Δ and such a solution trajectory $\vec{x}(t)$ is *continuous* and never leaves Δ.

4 SPP includes PPP

We now show that our model of Switching Probabilistic Protocols (SPP) is more general than the model of [1] in the sense that it can be used to define the Probabilistic Population Protocols (PPP). We do this by showing the following:

Theorem 1 Assume that a set of differential equations represent the continuous time dynamics of PPP as a limit of the discrete model. Then, the continuous time dynamics of SPP include those of PPP as a special case.

Proof. According to [1], the discrete-time dynamics of a Probabilistic Population Protocol (PPP) are given by a finite set of rules, R of the form

$$(p, q) \mapsto (p', q')$$

where $p, q, p', q' \in K$ ($K = \{q_1, \ldots, q_k\}$) together with a set A of n agents and an (irreflexive) relation $E \subseteq A \times A$.

Intruitively, a $(u, v) \in E$ means that u, v are able to interact. [1] assumes further that E consists of all ordered pairs of distinct elements from A.

A *population configuration* in [1] is a mapping $C : A \mapsto K$ (K is the set of states). Let C and C' be population configurations, and u, v be two distinct agents. [1] says that C can go to C' in one discrete step (denoted $C \overset{e}{\mapsto} C'$) via an *encounter* $e = (u, v)$ if

$$(C(u), C(v)) \mapsto (C'(u), C'(v))$$

is a rule in R. This means that the state $C(u)$ of u switches to $C'(u)$ and also $C(v)$ switches to $C'(v)$.

The execution of the system is defined to be a sequence $C_0, C_1, C_2, \ldots$ of configurations (where C_0 is the initial configuration) such that for each i, $C_i \mapsto C_{i+1}$. An execution is fair if for any C_i and C_j, such that $C_i \mapsto C_j$ and C_i occurs infinitely often in the execution, C_j also occurs infinitely often in the execution.

In the probabilistic version of the above, [1] further states that e (the ordered pair to interact) is chosen at random, independently and uniformly from all ordered pairs corresponding to edges e in $A \times A$ ([1] calls it the model of Conjugating Automata, inspired also by [4]).

Let us now assume that $n \to \infty$ and let $x_i = \lim_{n\to\infty} \frac{n_i}{n}$ be the population fraction at state $q_i \in K$ at a particular configuration C, at time t. Consider the rule ρ in R

$$(q_r, q_m) \mapsto (q_i, q_j)$$

Without loss of generality we assume in the sequel that $r \neq m$ and $i \neq j$ in such rules ρ in R. By the uniformity and randomness, the probability that such an e, that follows from rule ρ, is selected (as the encounter), is just $x_r(t)x_m(t)$. Let A_i be the set of all (r, m) that are the left part of a rule ρ:

$$\begin{array}{lccc} & (q_r, q_m) & \mapsto & (q_i, q_j) \\ \text{or} & (q_r, q_m) & \mapsto & (q_j, q_i) \end{array}$$

Let B_i be the set of (r, m) that are the left part of a rule ρ':

$$(q_r, q_m) \mapsto (q_{r'}, q_{m'})$$

with $r = i$ or $m = i$. Without loss of generality let $r = i$ in ρ'. By considering a small interval Δt and taking limits as $\Delta t \to 0$, due to fairness we get $\forall i$:

$$\dot{x_i} = \sum_{(r,m) \in A_i} x_r(t)x_m(t) - x_i(t) \sum_{(i,m) \in B_i} x_m(t) \qquad (2)$$

The above set of equations describe the continuous dynamics of PPP.

Now, consider our SPP dynamics and Eq. 1. Set $\lambda_i(\vec{x}) = \sum x_m(t)$, with m ranging over all rules

$$(q_r, q_m) \mapsto (q_{r'}, q_{m'})$$

with $r = i$, and all rules

$$(q_m, q_r) \mapsto (q_{r'}, q_{m'})$$

with $r = i$ (i.e., over all rules in B_i).

Also, set $p_{mi} = p_{ri} = 0$, if r, m do not belong in any tuple of A_i.

Finally set

$$p_{ri} = \frac{1}{\lambda_r} \sum_{m \in C(r,i)} x_m(t)$$

where $C(r,i)$ is the set of indices m in the second argument of the left part of rules in A_i (i.e. $(q_r, q_m) \mapsto (q_{r'}, q_{m'})$ with $r' = i$ or $m' = i$).

Then our system of Eq. 1 (the SPP dynamics) becomes the system of Eq. **??** (the PPP dynamics). Thus the PPP dynamics are a special case of the SPP dynamics in the continuous time setting. □

Here is an example of the reduction described above. Let the rules R in PPP be

$$\begin{aligned} (q_1, q_2) &\mapsto (q_3, q_2) \\ (q_3, q_1) &\mapsto (q_1, q_2) \\ (q_2, q_3) &\mapsto (q_2, q_1) \end{aligned}$$

This gives the continuous PPP dynamics:

$$\begin{aligned} \dot{x_1} &= x_1 x_3 + x_2 x_3 - x_1 (x_2 + x_3) \\ \dot{x_2} &= x_1 x_3 + x_1 x_2 + x_2 x_3 - x_2 (x_1 + x_3) \\ \dot{x_3} &= x_1 x_2 - x_3 (x_1 + x_2) \end{aligned}$$

We then set

$$\begin{aligned} \lambda_1 &= x_2 + x_3 \\ \lambda_2 &= x_1 + x_3 \\ \lambda_3 &= x_1 + x_2 \end{aligned}$$

and

$$\begin{array}{lll} p_{21} = \frac{x_3}{x_1+x_3} & p_{11} = \frac{x_3}{x_2+x_3} & p_{31} = 0 \\ p_{12} = \frac{x_3}{x_2+x_3} & p_{22} = \frac{x_1}{x_1+x_3} & p_{32} = \frac{x_2}{x_1+x_2} \\ p_{13} = \frac{x_2}{x_2+x_3} & p_{23} = p_{33} = 0 & \end{array}$$

and this results in our SPP dynamics, namely:

$$\begin{aligned} \dot{x_1} &= x_1 \lambda_1 p_{11} + x_2 \lambda_2 p_{21} + x_3 \lambda_3 p_{31} - x_1 \lambda_1 \\ \dot{x_2} &= x_1 \lambda_1 p_{12} + x_2 \lambda_2 p_{22} + x_3 \lambda_3 p_{32} - x_2 \lambda_2 \\ \dot{x_3} &= x_1 \lambda_1 p_{13} + x_2 \lambda_2 p_{23} + x_3 \lambda_3 p_{33} - x_3 \lambda_3 \end{aligned}$$

5 Stability of nonlinear dynamic systems: a sufficient condition for decidability.

Let us consider a dynamic system

$$\dot{x_i} = f_i(\vec{x}), \qquad i = 1, \ldots, k$$

that is, in fact, more general than Eq. 1.

Definition 1 (Fixed Points) Let $\vec{x}^*$ be a solution of the system $\{f_i(\vec{x}^*) = 0,\ i = 1, \ldots, k\}$ which we call a *fixed point* of the system.

By making a Taylor expansion around $\vec{x}^*$ we obtain a linear approximation to the dynamics:

$$\dot{x_i} = \sum (x_j - x_j^*) \frac{df_i}{dx_j} (\vec{x}^*)$$

Setting $\xi_i = x_i - x_i^*$ we get

$$\dot{\xi_i} = \sum \xi_j \frac{df_i}{dx_j} (\vec{x}^*)$$

which is a Linear System with a fixed point at the origin, i.e., $\dot{\xi} = L\xi$ where the matrix L has *constant* components $L_{ij} = \frac{df_i}{dx_j}(\vec{x}^*)$. L is called the Jacobian Matrix. Then, by the theorem of [5] we have

Corollary 2 If the fixed point $\vec{x}^*$ is *hyperbolic* (i.e., all eigenvalues of L^* have a non-zero real part) then the topology of the dynamics of the nonlinear system around $\vec{x}^*$ is the same as the topology of a $\vec{x}^*$ in the Linear system.

In fact, let each eigenvalue of L be $\phi = \mathsf{a} + i\omega$.

Corollary 3 Let $\mathsf{a} \neq 0, \forall \phi$ eigenvalues of L. Then

(a) If $\mathsf{a} < 0, \forall \phi$ then $\vec{x}(t)$ approaches the fixed point $\vec{x}^*$ as $t \to \infty$.

(b) If there exists a ϕ with $\mathsf{a} > 0$ then $\vec{x}(t)$ *diverges* from the fixed point $\vec{x}^*$ along the direction of the corresponding eigenvector. That is, the fixed point $\vec{x}^*$ is unstable.

Thus we get our main result of the system:

Theorem 4 If all fixed points $\vec{x}^*$ of our population dynamics of Eq. 1 are hyperbolic, then we *can decide stability* of the population protocol, around x^*, in *polynomial time* in the description of the protocol.

Corollary 5 If all fixed points of PPP are hyperbolic, then the stability of PPP can be decided in polynomial time.

6 Conclusions

The *population protocol* model of Angluin et. al. [1] consists of a (large) population of finite-state *agents* that interact in pairs. Each interaction updates the state of both participants according to a transition function based on the pair of the participants' previous states. A natural probabilistic model, proposed in [1], assumes each interaction to occur between a pair of agents chosen uniformly at random. We call the protocols of [1] by the term "Probabilistic Population Protocols" (PPP). [4] studied the acquisition and propagation of knowledge in the probabilistic model of random interactions between all paris in a population (conjugating automata). Curiously, the differential equation approach for such protocols was not proposed till now.

We imagine here a continuoum of agents. By the law of large numbers, one can model the underlying aggregate stochastic process as a deterministic flow system. Our main proposal here is to exploit the powerful tools of continuous nonlinear dynamics in order to examine questions (such as stability) of such protocols.

We have extended the class of [1] by defining a general model of "Switching Population Protocols" (SPP). Our main point is that one can study stability and population dynamics of protocols, via nonlinear differential equations that describe quite accurately the (discrete) population protocol dynamics when the population is very large. The "differential equations" approach was indicated in the past for the analysis of evolution of algorithms with Random Inputs, by [7, 8]. Our approach provides a sufficient condition for stability of PPP of [1] that can be checked in polynomial time. It also gives a more general way to *specify* population protocols, that reveals interesting classes.

References

[1] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. In *23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 290–299, New York, NY, USA, 2004. ACM.

[2] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.

[3] D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. In *20th International Symposium on Distributed Computing (DISC)*, volume 4167 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2006.

[4] Z. Diamadi and M. J. Fischer. A simple game for the study of trust in distributed systems. *Wuhan University Journal of Natural Sciences*, 6(1-2):72–82, March 2001. Also appears as Yale Technical Report TR-1207, January 2001, `ftp://ftp.cs.yale.edu/pub/TR/tr1207.ps`.

[5] P. Hartman. A lemma in the theory of structural stability of differential equations. *American Mathematical Society*, 11(4):610–620, 1960.

[6] M. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, 1974.

[7] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[8] N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5:1217–1235, 1995.