Interacting Particles Systems and Efficient Approximations for Large Sparse Graphs

Senior Thesis Advisor: Kavita Ramanan
Graduate Student Mentor: Ankan Ganguly
Mitchell Wortsman

Applied Mathematics, Brown University

2018
April

Abstract: *Stochastic dynamics arising in a variety of applications including epidemiology, statistical physics, and load balancing are modeled by so called interacting particle systems. The dynamics of interacting particle systems are governed by an underlying graph structure and are typically high dimensional systems involving a large number of particles. The high dimensionality makes the characterization of the dynamics of a typical particle challenging. Approximations for the dynamics have been mainly studied in the case where the underlying graph is dense. Building on recently developed theory, we present novel computational algorithms for approximating local dynamics on sparse graphs. We demonstrate the effectiveness of our approximations through numerical results, and show that in many cases they outperform existing algorithms.*

Acknowledgements: *Thank you to Professor Ramanan for showing me how to research, and for immeasurable patience and support throughout this process. Thank you to Ankan Ganguly for mentorship and guidance.*

Contents

1 Introduction

Three characteristics are ubiquitous in models of real world phenomena. Such models often include objects, interactions between objects, and uncertainty. An object can range from a single atom to a human being, but for the remainder of this work we will refer to an object abstractly as a particle. Accordingly, it is natural to study stochastic networks of particles, often called an interacting particle systems.

In specific interaction networks the dynamics of a typical particle is well understood. It was shown in [17] that as the number of particle tends to infinity we can use the so called mean field approximation to exactly characterize particle dynamics if the interaction network is complete. Moreover, it was recently shown in [4] that this approximation is asymptotically accurate for sequences of dense graphs whose degree approaches infinity. Consequently, the mean field approximation has become somewhat of the standard in industry and among the scientific community. However, this frequently used approximation is quite inaccurate for sparse graphs.

In this work we build on novel theoretical results by Lacker, Ramanan, and Wu in [11] to construct novel algorithms for approximating local dynamics on sparse graphs. We present an algorithm we call the local recursion, which is provably exact. However, our local recursion has exponential computational complexity in time and so we also give a polynomial time approximation.

This report is organized as follows. In Section 2 we present a formal definition of an interacting particle system along with a series of important examples. We then discuss existing approximations for the dynamics of a typical particle in Section 3. In Section 4 we use recently developed theory to construct our new local recursions. Finally we showcase the effectiveness of our algorithms in Section 5.

2 Interacting Particle Systems

When considering systems of many random variables it is natural to model the underlying interaction structure with a graph. In the study of probabilistic graphical models a single random variable is assigned to each node. However, in a myriad of applications including epidemiology, statistical physics, and load balancing, we are interested in the behavior of a system throughout time. We must therefore move beyond a static graphical model. Instead we consider each node to be a particle with a state that fluctuates throughout time.

2.1 Formal Definitions

2.1.1 Probabilistic Graphical Models

In the study of probabilistic graphical models we consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges. For each $i \in \mathcal{V}$ we let X_i be a random variable taking values in some countable space $\mathcal{X}$. It is standard notation to let $X = \{X_i\}_{i \in \mathcal{V}}$ and $X_A = \{X_i\}_{i \in A}$ for some $A \subseteq \mathcal{V}$.

It is often relevant to consider the boundary of a set $A \subseteq \mathcal{V}$, defined as

$$\partial A = \{v \in \mathcal{V} : (u, v) \in \mathcal{E} \text{ for some } u \in A\}. \tag{2.1}$$

We provide an example of A and ∂A for the lattice graph below.

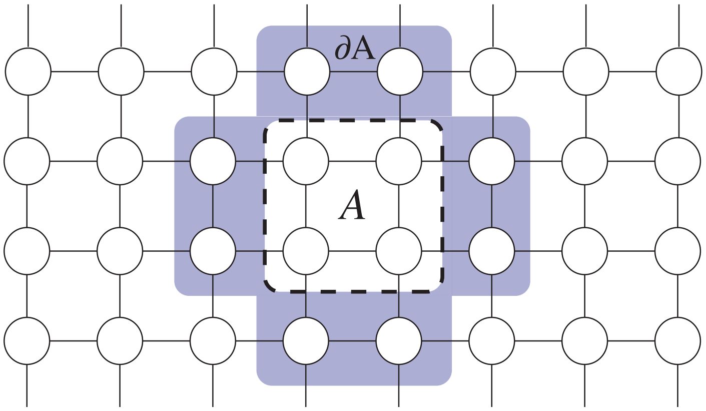

Figure 1: A and ∂A for a Lattice Graph ($\mathcal{V} = \mathbb{Z}^2$ and $\mathcal{E} = \{(u, v) : \|u - v\|_1 = 1\}$)

One of the most common classes of probabilistic graphical models is called a *Markov random field* (MRF), studied extensively in [10] . A probability measure $\mathbb{P}$ on X is said to be a MRF with respect to $\mathcal{G}$ if for any $A \subseteq \mathcal{V}$

$$\mathbb{P}\left(X_A = x_A | X_{\mathcal{V} \setminus A} = x_{\mathcal{V} \setminus A}\right) = \mathbb{P}\left(X_A = x_A | X_{\partial A} = x_{\partial A}\right) \tag{2.2}$$

for $x_{\mathcal{V}}$ such that $\mathbb{P}(X_{\mathcal{V}} = x_{\mathcal{V}}) > 0$. Equation 2.2 is often called the *spatial Markov property.*

In general for random variables X, Y and Z we say that X and Y are conditionally independent given Z if $\mathbb{P}(X = x|Y, Z) = \mathbb{P}(X = x|Z)$, or equivalently $\mathbb{P}(X = x, Y = y|Z) = \mathbb{P}(X = x|Z)\mathbb{P}(Y = y|Z)$. This is written as $X \perp\!\!\!\perp Y \mid Z$. Accordingly, we may restate the spatial Markov property as

$$X_A \perp\!\!\!\perp X_{\mathcal{V} \setminus (A \cup \partial A)} \mid X_{\partial A}. \tag{2.3}$$

Lemma 2.1. Consider a Markov Random Field $\mathbb{P}$ with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where A, B and S are a disjoint partition of $\mathcal{V}$. If every path

$$\{(u_0, u_1), (u_1, u_2), ..., (u_{k-1}, u_k)\} \subseteq \mathcal{E}$$

with $u_0 \in A, u_k \in B$ contains at least one node $u_j \in S$ then X_A is conditionally independent of X_B given X_S.

Proof. This follows immediately from $\partial A \cap B = \emptyset$. See Appendix for details. □

Lemma 2.1 is not immediately useful in this work. There is, however, an analogous Lemma for an interacting particle system by Lacker, Ramanan, and Wu in [11]. This theoretical result is fundamental for the development of our local recursions, as further discussed in Section 2.3.

2.1.2 Discrete Time Interacting Particle Systems

Interacting particle systems are modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of particles and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of pairwise interactions between particles. The state of particle $i \in \mathcal{V}$ at time $t \in \mathbb{T}$ is given by the random variable $X_i(t)$ which takes values in some countable space $\mathcal{X}$. $\mathbb{T} \subseteq \mathbb{R}^+ \cup \{0\}$ is a sequence of times which must be countable in the discrete time setting.

We let $X(t) = (X_i(t))_{i \in \mathcal{V}}$ denote the state of all particles at time t. Moreover we follow the notation of [9] to denote the trajectory of a particle up to time $T \in \mathbb{T}$ by a superscript as $X_i^T = (X_i(t))_{t \in \mathbb{T}, t \leq T}$. When $\mathbb{T} = \mathbb{N}$ we have $X_i^T = (X_i(0), X_i(1), ..., X_i(T))$.

Figures 2 and 3 illustrate two possible interaction networks.

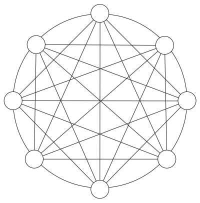

Figure 2: Complete Interaction Network

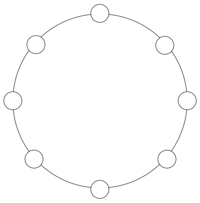

Figure 3: Ring Interaction Network

There are three properties satisfied by the interacting particle systems we consider.

1. An interacting particle system must obey the *Markov property.* A probability measure $\mathbb{P}$ on $\{X(t) : t \in \mathbb{T}\}$ is said to satisfy the Markov property if

$$X(t) \perp\!\!\!\perp \{X(s') : s' < s\} \mid X(s) \tag{2.4}$$

for any $s, t \in \mathbb{T}$ with $s < t$.

2. In an interacting particle system, each particle changes its state according only to its present state and the state of its neighbors. We call this the *local interactions property.*

   Formally, there exists some measurable function g such that

   $$\mathbb{P}(X_i(t) = v \mid X_i(t-1) = u) = g(v, u, X_{\partial i}(t-1)). \tag{2.5}$$

   We may equivalently introduce independent random variables $\xi = \{\xi_i(t)\}_{i \in \mathcal{V}, t \in \mathbb{T}}$ and let

   $$X_i(t) = f(X_i(t-1), X_{\partial i}(t-1), \xi_i(t)) \tag{2.6}$$

   for some measurable function f. The random variables $\xi_i(t)$ introduce randomness. Without ξ the system is deterministic given the initial conditions $X(0)$. It is important to note that $\xi_i(t)$ is independent not only of every $\xi_j(s)$ but also $X_j(s)$ for $(j, s) \neq (i, t)$.

   Remark 2.1. The formulations given by equations 2.5 and 2.6 are equivalent. Proof of this equivalence may be found in the appendix.

3. The initial conditions $\{X_u(0)\}_{u \in \mathcal{V}}$ are independent. Though this property need not always hold, it will make our analysis simpler.

2.1.3 Continuous Time Interacting Particle Systems

The extension to an uncountable set $\mathbb{T} \subseteq \mathbb{R}^+ \cup \{0\}$ is quite natural. The only additional restriction is that $X_i(t)$ is right continuous and has finite left limits with respect to t. The Markov property in equation 2.4 is unchanged and equation 2.6 simply becomes

$$X_i(t) = f(X_i(t-), X_{\partial i}(t-), \xi_i(t)) \tag{2.7}$$

where $X_A(t-) = \lim_{s \nearrow t} X_A(s)$ for any A.

2.2 Examples

In this section we present some interesting and relevant examples of interacting particle systems. We begin with a discussion of compartmental models in epidemiology then move to load balancing and classical examples from statistical physics. We are brief in our description as the study of each model could comprise an entire report.

2.2.1 The SIR Process

Proposed by Kermack and McKendrick in 1927, the susceptible-infected-recovered (SIR) process was one of the first models for epidemics [15]. In the SIR process a particle can transition from susceptible to infected and from infected to recovered. Though the model was originally developed to study the spread of disease among human beings, it has recently found various applications in new fields including distributed systems.

Traditionally the SIR process is studied in the case where every particle interacts with every other particle. The system is then expressed as the series of differential equations

$$\begin{cases} \frac{dS_N}{dt} = -\beta I_N S_N \\ \frac{dI_N}{dt} = \beta I_N S_N - \alpha I_N \\ \frac{dR_N}{dt} = \alpha R_N. \end{cases} \tag{2.8}$$

where S_N is the number of susceptible particles, I_N is the number of infected particles, and R_N is the number of recovered particles. A particle recovers at rate α and infection occurs at rate βI_N. For the purpose of this report we will be considering an analogous formulation where the infection rate scales with the number of neighbors.

We now present the analogous discrete time process we will consider. For an interaction network $\mathcal{G}$ where each node has exactly d neighbors, we consider the SIR process where the state of a particle at time t is either S, I or R. The system may be described by the following equations

$$\begin{cases} \{X_u(0)\}_{u\in\mathcal{V}} \sim \text{i.i.d. } \nu \\ \mathbb{P}(X_i(t) = I | X_i(t-1) = S) = \frac{p}{d}\sum_{j\in\partial i} \mathbb{1}_{\{X_j(t-1)=I\}} \\ \mathbb{P}(X_i(t) = R | X_i(t-1) = I) = q \\ \mathbb{P}(X_i(t) = S | X_i(t-1) = I) = \mathbb{P}(X_i(t) = I | X_i(t-1) = R) = 0 \end{cases} \tag{2.9}$$

where p/d is the equivalent of β and q is analogous to α. We visualize the transition dynamics in figure 4.

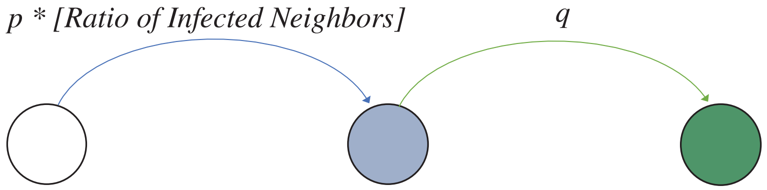

Figure 4: Dynamics of Susceptible (white), Infected (blue), and Recovered (green).

Figures 5 and 6 showcase simulations of the SIR process on two different interaction networks with 40 particles and parameters $p = 0.5$, $q = 0.1$. The top plot displays the number of particles in each state while the bottom plot shows the states at the level of an individual particle. On the bottom plot, each row is a different particle with the susceptible state shown in white, infected in blue, and recovered in green. Note that local bands of infection are prevalent when the interaction network is a ring.

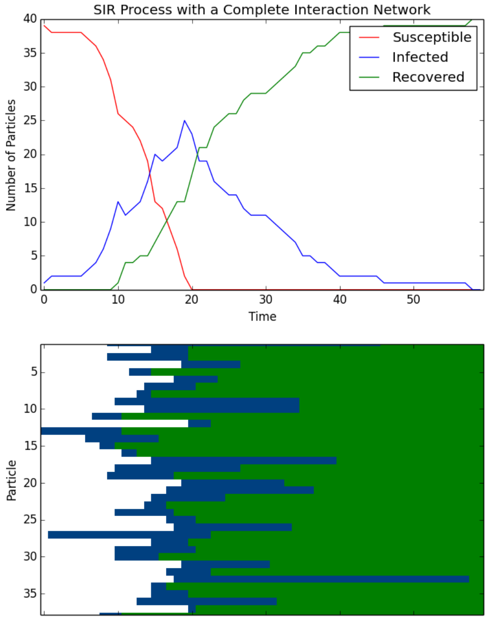

Figure 5: Sir Process (Complete)

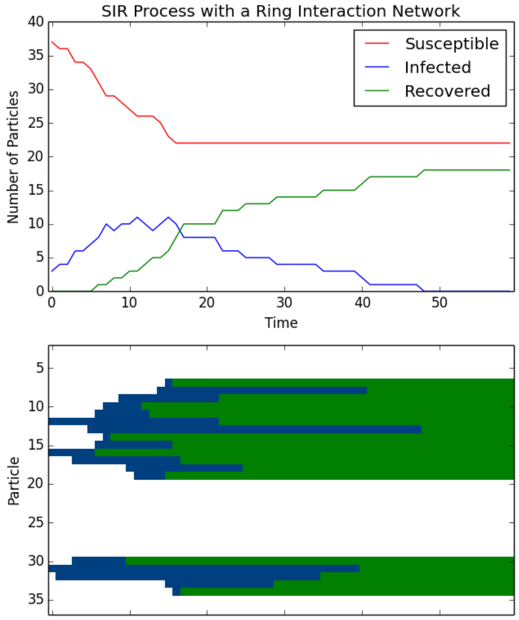

Figure 6: SIR Process (Ring)

2.2.2 The Contact Process

The contact process is another model for the spread of disease. It is similar to the SIR process but has no recovered state. A particle instead transitions back to susceptible. The contact process is studied extensively on trees in [19]. Formally, $X_i(t)$ is either 0 (susceptible) or 1 (infected) and the discrete transition probabilities may be described by the following equations.

$$\begin{cases} \{X_u(0)\}_{u\in\mathcal{V}} \sim \text{i.i.d. } \nu \\ \mathbb{P}(X_i(t) = 1 \mid X_i(t-1) = 0) = \frac{p}{d}\sum_{j\in\partial i} X_j(t-1) \\ \mathbb{P}(X_i(t) = 0 \mid X_i(t-1) = 1) = q \end{cases} \tag{2.10}$$

For the contact process we also give the continuous time formulation. We let N_i be a poisson process of rate q for all $i \in \mathcal{V}$ and let $N_{i,j}$ be a poisson process of rate p/d

for $i, j \in \mathcal{E}$. The contact process in continuous time is then described by

$$\begin{cases} \{X_u(0)\}_{u \in \mathcal{V}} \sim \text{i.i.d. } \nu \\ X_i(t) = X_i(0) - \int_0^t X_i(s) N_i(ds) + \sum_{j \in \partial i} \int_0^t (1 - X_j(s)) X_i(s) N_{i,j}(ds) \end{cases} \tag{2.11}$$

where

$$\int_0^t X_i(s) N_i(ds) \tag{2.12}$$

is the number of times particle i recovers and

$$\sum_{j \in \partial i} \int_0^t (1 - X_j(s)) X_i(s) N_{i,j}(ds) \tag{2.13}$$

is the number of times particle i becomes infected.

2.2.3 Load Balancing

If we wish to understand how network traffic should be allocated it is imperative to understand how servers behave given an allocation scheme. This problem has become increasingly important with the advent of distributed technology such as the cloud. The system studied by Mitzenmacher in [16] consists of a network of N servers, each with a queue of jobs. Job arrivals are modeled by a Poisson stream with rate $N\lambda$ for $(\lambda < 1)$ and the processing time for each job is exponentially distributed with mean 1. For each incoming job, d servers are chosen at random and the job is allocated to the server with the shortest queue. If $X_i(t)$ denotes the number of jobs in the queue of server i at time t, then we are interested in the fixed point of $s_k = \mathbb{P}(X_i(t) > k)$ as $t, N \to \infty$. It is well known that for $d = 1$, s_k decreases exponentially in k. However, Mitzenmacher demonstrates in [16, Lemma 2] that for $d > 1$ the system converges to the unique fixed point

$$s_k = \lambda^{\frac{d^i - 1}{d - 1}} \tag{2.14}$$

and so s_k is said to decrease doubly exponentially for $d \geq 1$.

The paradigm studied by Gast in [6] is similar, but accounts for the underlying graph structure of the server network $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$. For each incoming job, a server u_1 is chosen uniformly from $\mathcal{V}_s$ then a second server u_2 is chosen uniformly from ∂u_1. The job is then allocated to the server among u_1 and u_2 with the shortest queue. When $\mathcal{G}$ is the complete graph this model corresponds exactly to that of Mitzenmacher for $d = 2$.

We now consider a similar model in which jobs arrive at each of the N servers as a Poisson stream of rate λ. A job incident at server j is allocated to the server with the shortest queue among j and ∂j. An arrival at j which is allocated to $k \in \partial j$ is shown in Figure 7.

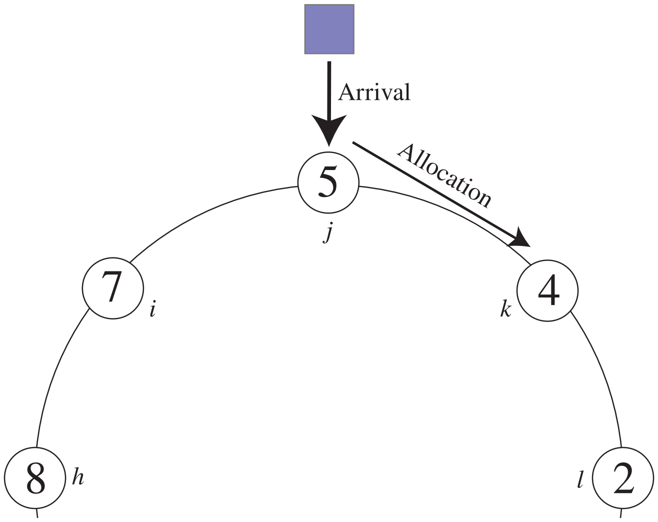

Figure 7: An Arrival at Server j is Allocated to Server k

When we formulate the problem as an interacting particle system the *interaction network* differs from the *server network*, as illustrated by an example in figures 8 and 9 for load balancing on the ring.

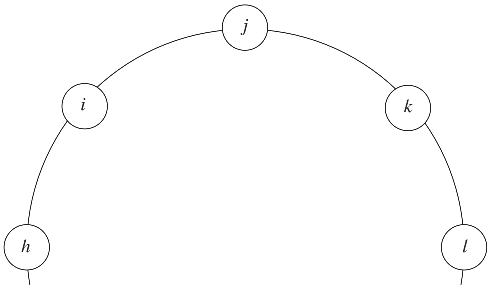

Figure 8: Server Network $\mathcal{G}_s$

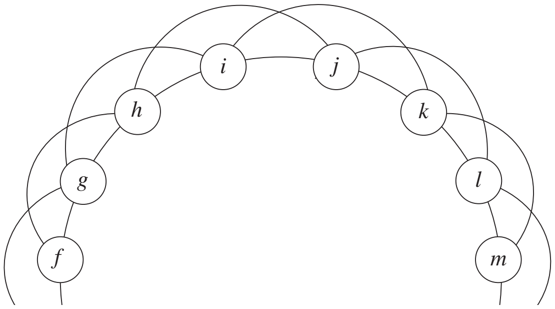

Figure 9: Interaction Network $\mathcal{G}_i$

We refer back to figure 7 for an example which illustrates the discrepancy between the networks. We may not claim that the state of server j at time t evolves based only upon the states of servers i and k. Consider an arrival at server i, an event which may change the state of server j. Clearly we must know the state of server h in order to determine if the job which has arrived at i will be routed to server j.

2.2.4 Ising

The Ising model is a mathematical model for ferromagnetism in which each particle has magnetic spin up ($+1$) or down (-1). Developed in 1921 by Wilhelm Lenz, the model was later named after his student Ernst Ising. The Ising model is discussed further in [12, 13] and we provide some relevant theory below.

The Ising model consists of N particles with an interaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that is traditionally a lattice. For each configuration of particle spins $\sigma \in \{-1, 1\}^N$ there is an associated energy $H(\sigma)$. A probability distribution π over all possible configurations may be obtained by enforcing that lower energy configurations are more likely to be observed. For $\beta > 0$ and normalization constant $Z_{\beta,n}$, $\pi(\sigma)$ is given by

$$\pi(\sigma) = \frac{1}{Z_{\beta,N}} e^{-\beta H(\sigma)} \tag{2.15}$$

where

$$Z_{\beta,N} = \sum_{\sigma \in \{-1,1\}^N} e^{-\beta H(\sigma)}. \tag{2.16}$$

As $\beta \to 0$ all configurations become equally likely. Alternatively, when β is large, π is concentrated at low energy configurations.

The energy function H, also be referred to as *the Hamiltonian*, depends on parameters $J, h \in \mathbb{R}$.

$$H(\sigma) = -J \sum_{(u,v) \in \mathcal{E}} \sigma_u \sigma_v - h \sum_{u \in \mathcal{V}} \sigma_u. \tag{2.17}$$

The first term of equation 2.17 is referred to as the *interaction energy* while the second term is called the *external field*. When $J > 0$ the system is said to be *ferromagnetic* as the interaction energy will be lower when neighboring particles have matching spins. The second term accounts for an external magnetic field which breaks the symmetry between positive and negative spins. When $h > 0$ we expect configurations with more positive spin particles to be more likely.

It is important to note that π, which may also be referred to as a *Gibbs Measure*, obeys the spatial Markov property with respect to the interaction network $\mathcal{G}$.

For a general interaction network it is computationally infeasible to compute π. However, we may use Markov Chain Monte Carlo (MCMC) techniques to obtain a sample σ from π. We describe one variant of MCMC for Ising in Algorithm 1. Though beyond the scope of this report there is a myriad of theory surrounding the exact number of iterations required for algorithm 1, which we denote S.

Algorithm 1 IsingMCMC

```
Initialize $\sigma_i$ i.i.d. for $i \in \mathcal{V}$.
for $s = 1, 2, ..., S$ do
    Choose $k \in \{1, ..., N\}$ uniformly.
    Let $\tau_j = \sigma_j$ for all $j \neq k$ and $\tau_k = 1 - \sigma_k$.
    Let $\sigma = \tau$ with probability $\min\{1, \pi(\tau)/\pi(\sigma)\}$.
end for
return $\sigma$
```

Algorithm 1 relies on the fact that we may calculate the ratio $\pi(\tau)/\pi(\sigma)$ without explicitly computing π.

$$\frac{\pi(\tau)}{\pi(\sigma)} = e^{-\beta(H(\tau)-H(\sigma))} \tag{2.18}$$

and so when $\tau_j = \sigma_j$ for all $j \neq k$ we have

$$H(\tau) - H(\sigma) = -\sum_{(k,v)\in\mathcal{E}} J\tau_k\sigma_v - h\tau_k - \left(-\sum_{(k,v)\in\mathcal{E}} J\tau_k\sigma_v - h\sigma_k\right) \tag{2.19}$$

$$= -\sum_{v\in\partial k} \frac{J\tau_k\sigma_v}{2} - \frac{h\tau_k}{2} \tag{2.20}$$

as $\tau_k - \sigma_k = \tau_k/2$.

We now verify that as $S \to \infty$, algorithm 1 returns a sample σ from π. Let $P(\sigma, \tau)$ be the probability of transitioning from configuration σ to configuration τ in the body of the **for** loop. By examining algorithm 1 we find that P is given by

$$P(\sigma,\tau) = \begin{cases} \frac{1}{N}\min\{1, \pi(\tau)/\pi(\sigma)\} & \text{if } \|\tau - \sigma\|_1 = 2 \\ 0 & \text{if } \|\tau - \sigma\|_1 > 2 \\ 1 - \sum_{\tau'\neq\sigma} P(\sigma,\tau') & \text{if } \tau = \sigma. \end{cases} \tag{2.21}$$

We may restate algorithm 1 as the process of choosing $\sigma^{(0)}$ at random then drawing $\sigma^{(n)}$ from the distribution $\mu_n(\sigma^{(n)}) = P\left(\sigma^{(n-1)}, \sigma^{(n)}\right)$ for $1 \leq n \leq T$.

We now claim that the Markov chain generated by matrix P has stationary distribution π. This will imply that as S grows large σ is distributed according to π. A sufficient condition for P to have stationary distribution π is

$$\pi(\sigma)P(\sigma,\tau) = \pi(\tau)P(\tau,\sigma). \tag{2.22}$$

We will now show that equation 2.22 holds. Without loss of generality assume $\pi(\tau) < \pi(\sigma)$ and observe that

$$\pi(\sigma)P(\sigma,\tau) = \pi(\sigma)\frac{1}{N}\min\{1, \pi(\tau)/\pi(\sigma)\} \tag{2.23}$$

$$= \pi(\sigma)\frac{1}{N}\frac{\pi(\tau)}{\pi(\sigma)} \tag{2.24}$$

$$= \pi(\tau)\frac{1}{N} \tag{2.25}$$

$$= \pi(\tau)\frac{1}{N}\min\{1, \pi(\sigma)/\pi(\tau)\} \tag{2.26}$$

$$= \pi(\tau)P(\tau, \sigma). \tag{2.27}$$

We now introduce an analogous discrete time interacting particle system, as studied in [2] and [13], which we refer to as *parallel Ising.*

If a particle k is chosen to switch its value in the body of the **for** loop, the transition from σ_k to τ_k occurs with probability

$$\min\left\{\exp\left(-\beta\left(-\sum_{(k,v)\in\mathcal{E}}\frac{J\tau_k\sigma_v}{2} - \frac{h\tau_k}{2}\right)\right),\ 1\right\} \tag{2.28}$$

which depends only on particles ∂k and k. Accordingly, we may construct an interacting particle system which approximates algorithm 1 as follows. Let $X_i(0)$ be chosen i.i.d. as in algorithm 1. Then for $t \geq 1$ and $i \in \mathcal{V}$ let the transition probability be

$$\begin{aligned}&\mathbb{P}\left(X_i(t) = x_i(t) \mid X_i(t-1) = x_i(t-1)\right)\\ &= \frac{1}{N}\min\left\{\exp\left(-\beta\left(-\sum_{j\in\partial i}\frac{Jx_i(t)X_j(t-1)}{2} - \frac{hx_i(t)}{2}\right)\right),\ 1\right\}\end{aligned} \tag{2.29}$$

for $x_i(t) \neq x_i(t-1)$. We may then account for the case of $x_i(t) = x_i(t-1)$ as the total probability must sum to 1. It is important to note that the dynamics of *parallel Ising* are not exactly identical to algorithm 1 as *parallel Ising* allows for the value of multiple particles to flip from one time-step to the next.

2.2.5 Potts

Introduced by Renfrey Potts in 1951, the Potts model generalizes the Ising model to multiple states. Instead of having spin up (+1) or down (-1) a particle may have spin $\{1, 2, ..., q\}$ for $q \in \mathbb{N}$. As in Ising, the value of the Hamiltonian fluctuates based on how many neighboring particle pairs have matching spins. In the Potts model the Hamiltonian H becomes

$$H(\sigma) = -J\sum_{(u,v)\in\mathcal{E}}\delta(\sigma_u, \sigma_v) - h\sum_{u\in\mathcal{V}}\sigma_u \tag{2.30}$$

where $\delta(\sigma_u, \sigma_v) = 1$ if $\sigma_u = \sigma_v$ and 0 otherwise. The distribution π given by equation 2.15 is unchanged and only a small modification is made to algorithm 1.

Algorithm 2 PottsMCMC

```
Initialize $\sigma_i$ i.i.d. for $i \in \mathcal{V}$.
for $s = 1, 2, ..., S$ do
    Choose $k \in \{1, ..., N\}$ uniformly.
    Choose $\ell$ uniformly from from $\{1, ..., q\} \setminus \{\sigma_k\}$.
    Let $\tau_j = \sigma_j$ for all $j \neq k$ and $\tau_k = \ell$.
    Let $\sigma = \tau$ with probability $\min\{1, \pi(\tau)/\pi(\sigma)\}$.
end for
return $\sigma$
```

In the Potts model it is still feasible to compute $\pi(\tau)/\pi(\sigma) = e^{-\beta(H(\tau)-H(\sigma))}$ when $\tau_j = \sigma_j$ for all $j \neq k$ as

$$H(\tau) - H(\sigma) = -J \sum_{v \in \partial k} \left(\delta(\sigma_u, \tau_k) - \delta(\sigma_u, \sigma_k)\right) - h\left(\tau_k - \sigma_k\right). \tag{2.31}$$

The interacting particle system analogous to algorithm 2, which we denote *parallel Potts*, is nearly identical to 2.29. The transition probabilities for $x_i(t) \neq x_i(t-1)$ are now

$$\mathbb{P}\left(X_i(t) = x_i(t) \mid X_i(t-1) = x_i(t-1)\right) = \frac{1}{N(q-1)} \min\left\{\exp\left(-\beta H^*\right),\ 1\right\} \tag{2.32}$$

where

$$\begin{aligned} H^* = &- J \sum_{j \in \partial i} \delta(X_j(t-1), x_i(t)) - \delta(X_j(t-1), x_i(t-1)) \\ &- h\left(x_i(t) - x_i(t-1)\right). \end{aligned} \tag{2.33}$$

We may then account for the case of $x_i(t) = x_i(t-1)$ since the total probability must sum to 1.

2.3 Conditional Independence and the Double Boundary

Lacker, Ramanan, and Wu present in [11] the theoretical result which is the cornerstone for the development of our local recursions. They show that for an interacting particle system with interaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and any set $A \subseteq \mathcal{V}$, X_A^T is conditionally independent of all other particles given its *double boundary*. Formally,

$$X_A^T \perp\!\!\!\perp X_{\mathcal{V}\setminus(A \cup \partial^2 A)}^T \mid X_{\partial^2 A}^T. \tag{2.34}$$

where $X_A^T = \left(X_A(t)\right)_{t \leq T, t \in \mathbb{T}}$ denotes the full trajectory of A and $\partial^2 A$ is the *double boundary* of A. The *double boundary* includes not only the neighbors of A but also the neighbors of the neighbors, defined as

$$\partial^2 A = \{v \in \mathcal{V} \setminus A\ :\ \text{there exists a path from } u \in A \text{ to } v \text{ of length 1 or 2}\} \tag{2.35}$$

and is illustrated by figure 10.

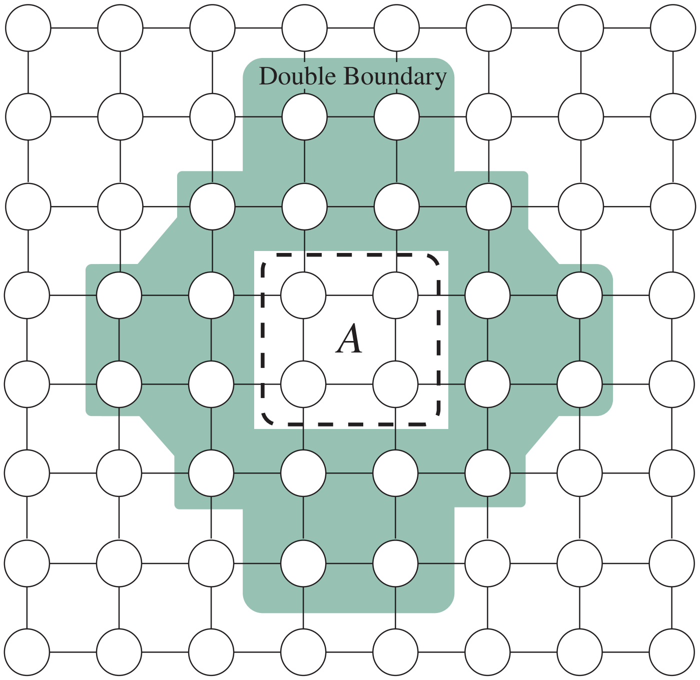

Figure 10: The Double Boundary $\partial^2 A$ on a Lattice Graph

It is shown in [11] that equation 2.34 holds in discrete time, continuous time, and for diffusions. It is also shown that 2.34 holds even for non-finite interaction networks. Though a full proof is beyond the scope of this report, we may follow their techniques to provide a proof in the discrete time case where $\mathbb{T} = \mathbb{N}$ and the interaction graph is finite.

For the remainder of this section we assume $\mathbb{T} = \mathbb{N}$ and $|\mathcal{V}| < \infty$. We begin with the remark that in general, interacting particle systems need not obey the spatial Markov property, which requires conditional independence given the single boundary.

Lemma 2.2. For a general interacting particle system, both the time marginals $X_i(t)$ and the complete trajectories $X_i^T = (X_i(0), X_i(1), ..., X_i(T))$ need not obey the spatial Markov property introduced in equation 2.2.

Proof. A full and detailed counterexample may be found in the appendix, though we present a short argument here to provide intuition.

Consider four adjacent particles i, j, k, l on a line graph with as depicted by figure 11.

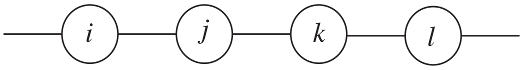

Figure 11
Four Particles on a Line Graph

Now consider the *exclusion process* on a line graph. If particle b has left neighbor a and right neighbor c then $X_b(t) = \mathrm{XOR}(X_a(t-1), X_c(t-1))$. Moreover we consider initial conditions $\{X_v(0)\}_{v\in\mathcal{V}} \sim$ i.i.d. Bernoulli(0.5). The spatial Markov property would imply that X_j^T is conditionally independent of X_l^T given X_i^T, X_k^T. However, if we know that $X_j(0) = 0$ and $X_k(1) = 1$ then it must be true that $X_l(0) = 1$, and so we do not have conditional independence. □

We now discuss the joint distribution of $X^T = (X_i(t))_{i\in\mathcal{V}, t\le T}$.

Lemma 2.3. For any discrete time, finite interacting particle system with independent initial conditions,

$$\mathbb{P}(X^T = x^T) = \prod_{i\in\mathcal{V}} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right) \tag{2.36}$$

for some set of functions $\{\mu_i^t\}_{i\in\mathcal{V}, t\in\mathbb{T}}$.

Proof. From the Markov property we may immediately obtain the factorization

$$\mathbb{P}(X^T = x^T) = \mathbb{P}(X(0) = x(0)) \prod_{t=1}^{T} \mathbb{P}\left(X(t) = x(t) \mid X(t-1) = x(t-1)\right) \tag{2.37}$$

and using the assumed independence of $\{X_i(0)\}_{i\in\mathcal{V}}$ we may write

$$\mathbb{P}(X(0) = x(0)) = \prod_{i\in\mathcal{V}} \mu_i^0(x_i(0)) \tag{2.38}$$

where $\mu_i^0(x_i(0)) = \mathbb{P}(X_i(0) = x_i(0))$. We now focus our attention on the term

$$\mathbb{P}\left(X(t) = x(t) \mid X(t-1) = x(t-1)\right). \tag{2.39}$$

In order to simplify equation 2.39 we first show that

$$X_i(t) \perp\!\!\!\perp X_j(t) \mid X(t-1). \tag{2.40}$$

Recall that $X_i(t) = f(X_i(t-1), X_{\partial i}(t-1), \xi_i(t))$ for some measurable function f taking values in a finite space and therefore

$$\mathbb{P}(X_i(t) = u \mid X(t-1) = x) = \mathbb{P}(\xi_i(t) \in \Delta(u, x_i, x_{\partial i})) \tag{2.41}$$

for $\Delta(u, x_i, x_{\partial i})$ defined as

$$\Delta(u, x_i, x_{\partial i}) = \{r \in \mathbb{R} : f(x_i, x_{\partial_i}, r) = u\} \tag{2.42}$$

It follows that

$$\begin{aligned}
&\mathbb{P}\left(X_i(t) = u, X_j(t) = v \mid X(t-1) = x\right) && (2.43)\\
&= \mathbb{P}(\xi_i(t) \in \Delta(u, x_i, x_{\partial i}),\ \xi_j(t) \in \Delta(v, x_j, x_{\partial j})) && (2.44)\\
&= \mathbb{P}(\xi_i(t) \in \Delta(u, x_i, x_{\partial i}))\mathbb{P}(\xi_j(t) \in \Delta(v, x_j, x_{\partial j})) && (2.45)\\
&= \mathbb{P}(X_i(t) = u \mid X(t-1) = x)\mathbb{P}(X_j(t) = v \mid X(t-1) = x) && (2.46)
\end{aligned}$$

and so $X_i(t) \perp\!\!\!\perp X_j(t) \mid X(t-1)$. In 2.45 we use that $\xi_i(t)$ and $\xi_j(t)$ are independent. Finally, we note that

$$\begin{aligned}
&\mathbb{P}(X_i(t) = u \mid X(t-1) = x) && (2.47)\\
&= \mathbb{P}(\xi_i(t) \in \Delta(u, x_i, x_{\partial i})) && (2.48)\\
&= \mathbb{P}(X_i(t) = u \mid X_i(t-1) = x_i, X_{\partial i}(t-1) = x_{\partial i}) && (2.49)
\end{aligned}$$

which implies that $X_i(t) \perp\!\!\!\perp X_{\mathcal{V}\setminus(i\cup\partial i)}(t-1) \mid X_{\partial i}(t-1)$. We may then define

$$\mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) = \mathbb{P}\left(\xi_i(t) \in \Delta(x_i(t), x_i(t-1), x_{\partial i}(t-1))\right) \tag{2.50}$$

and conclude that

$$\mathbb{P}(X^T = x^T) = \prod_{i\in\mathcal{V}} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1))\right). \tag{2.51}$$

□

Using the factorization above we are now able to prove a special case of equation 2.34.

Lemma 2.4. Equation 2.34 holds for any discrete time, finite interacting particle system with independent initial conditions. Equivalently, for any $A \subseteq \mathcal{V}$ and configuration x^T with non-zero probability,

$$\mathbb{P}(X_A^T = x_A^T | X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T) = \mathbb{P}(X_A^T = x_A^T | X_{\partial^2 A}^T = x_{\partial^2 A}^T). \tag{2.52}$$

Proof. Using that $\mathbb{P}(X_A^T = x_A^T) > 0$ we write

$$\begin{aligned}
&\mathbb{P}(X^T = x^T | X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T) && (2.53)\\
&= \frac{\mathbb{P}(X_A^T = x_A^T, X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T)}{\sum_{y_A^T} \mathbb{P}(X_A^T = y_A^T, X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T)} && (2.54)
\end{aligned}$$

For simplicity we will rewrite equation 2.54 as

$$\frac{\mathbb{P}(X_A^T = x_A^T, X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T)}{\sum_{x_A^T} \mathbb{P}(X_A^T = x_A^T, X_{\mathcal{V}\setminus A}^T = x_{\mathcal{V}\setminus A}^T)} \tag{2.55}$$

where it is assumed that in the denominator all other terms are held fix but the term being summed out.

We now return to the factorization given in Lemma 2.3 which we may apply to both the numerator and denominator in equation 2.55. However, we first note that we may extend Lemma 2.3 to further to decompose $\mathbb{P}(X^T = x^T)$ into two terms as

$$\begin{aligned}
&\prod_{i\in\mathcal{V}} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right) \\
&= \prod_{i\in A\cup\partial A} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right) \\
&\quad \prod_{i\in\mathcal{V}\setminus(A\cup\partial A)} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right)
\end{aligned} \tag{2.56}$$

The latter term in equation 2.56 does not depend on x_A^T and so can be factored out of the summation in the denominator and cancelled out from the numerator of equation 2.55. We may then rewrite equation 2.55 as

$$\frac{\prod_{i\in A\cup\partial A} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right)}{\sum_{x_A^T} \prod_{i\in A\cup\partial A} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right)} \tag{2.57}$$

Finally, we multiply both sides of equation 2.57 by

$$\sum_{x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T} \prod_{i\in\mathcal{V}\setminus(A\cup\partial A)} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right) \tag{2.58}$$

and we may bring the product $\prod_{i\in A\cup\partial A}(...)$ inside the sum as it does not contain any terms which depend on $x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T$. Equation 2.57 then simplifies to

$$\frac{\sum_{x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T} \prod_{i\in\mathcal{V}} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right)}{\sum_{x_A^T} \sum_{x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T} \prod_{i\in\mathcal{V}} \left(\mu_i^0(x_i(0)) \prod_{t=1}^{T} \mu_i^t(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right)} \tag{2.59}$$

$$= \frac{\sum_{x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T} \mathbb{P}\left(X_A^T = x_A^T, X_{\partial^2 A}^T = x_{\partial^2 A}^T, X_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T = x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T \right)}{\sum_{x_A^T} \sum_{x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T} \mathbb{P}\left(X_A^T = x_A^T, X_{\partial^2 A}^T = x_{\partial^2 A}^T, X_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T = x_{\mathcal{V}\setminus(A\cup\partial^2 A)}^T \right)} \tag{2.60}$$

$$= \frac{\mathbb{P}(X_A^T = x_A^T, X_{\partial^2 A}^T = x_{\partial^2 A}^T)}{\mathbb{P}(X_{\partial^2 A}^T = x_{\partial^2 A}^T)} = \mathbb{P}(X_A^T = x_A^T | X_{\partial^2 A}^T = x_{\partial^2 A}^T) \tag{2.61}$$

as needed. □

We conclude this section with a Lemma for interacting particle systems analogous to Lemma 2.1 for graphical models.

Lemma 2.5. Consider an interacting particle system with interaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where A, B and S are a disjoint partition of $\mathcal{V}$. If every path

$$\{(u_0, u_1), (u_1, u_2), ..., (u_{k-1}, u_k)\} \subseteq \mathcal{E}$$

with $u_0 \in A, u_k \in B$ contains at least two particles $u_j \in S$ then X_A^T is conditionally independent of X_B^T given X_S^T.

Proof. This follows immediately from $\partial^2 A \cap B = \emptyset$ and is nearly equivalent to the proof of Lemma 2.1. See Appendix for details. □

3 Existing Approximations

In this section we discuss existing approximations for characterizing particle dynamics.

3.1 Mean Field Approximation

3.1.1 Description

The so called mean field approximation is widely used for understanding the dynamics of a typical particle. In 1984 it was shown by Oelschlger in [17] that this approximation is exact as the population size tends to infinity if the interaction network is complete. More recently in [4] it was shown that the mean field approximation is asymptotically accurate for sequences of dense graphs where the degree of each particle tends to infinity. However, the mean field approximation is fairly inaccurate when the interaction network is sparse.

In the mean field approximation it is assumed that the neighbors of particle i are independent copies of i. Recall that for a discrete time interacting particle system, particle i evolves based on its own state and the state of its neighbors as

$$X_i(t) = f(X_i(t-1), X_{\partial i}(t-1), \xi_i(t)). \tag{3.1}$$

However, in the mean field approximation particle i updates according to its own state and its Law as

$$X_i(t) = \tilde{f}(X_i(t-1), \mathrm{Law}(X_i(t-1)), \xi_i(t)) \tag{3.2}$$

where the neighbors of i at time t are approximated by independent samples from $\text{Law}(X_i(t-1))$. Formally if i has d neighbors then

$$\begin{aligned} &\tilde{f}(X_i(t-1), \text{Law}(X_i(t-1)), \xi_i(t)) \\ &= f(X_i(t-1), \left[Y_1, ..., Y_d\right], \xi_i(t)) \end{aligned} \tag{3.3}$$

where $\{Y_i\}_{i=1}^d$ are i.i.d $\text{Law}(X_i(t-1))$.

In continuous time equation 3.2 simply becomes

$$X_i(t) = \tilde{f}(X_i(t-), \text{Law}(X_i(t-)), \xi_i(t)). \tag{3.4}$$

An analogous formulation of the mean field approximation as given by equation 3.2 is presented below.

$$\mathbb{P}(X_i(t) = v \mid X_i(t-1) = u) = \tilde{Q}(v, u, \text{Law}(X_i(t-1))) \tag{3.5}$$

As before we have that $\tilde{Q}$ implicitly assumes that the neighbors of i at t and independent samples from $\text{Law}(X_i(t))$. The equivalence of equations 3.2 and 3.5 follows from Remark 2.1 in which g is introduced. Formally, if i has d neighbors then

$$\tilde{Q}(v, u, \text{Law}(X_i(t-1))) = g(u, v, \left[Y_1, ..., Y_d\right]) \tag{3.6}$$

where $\{Y_i\}_{i=1}^d$ are i.i.d $\text{Law}(X_i(t-1))$.

Recall that in the contact process described by Section 2.2.2 a particle i transitions from Susceptible (0) to Infected (1) based on the ratio of infected neighbors. In discrete time we had

$$\mathbb{P}(X_i(t) = 1 \mid X_i(t-1) = 0) = \frac{1}{d} \sum_{j \in \partial i} X_j(t-1). \tag{3.7}$$

When $X_j(t-1)$ is approximated by an independent sample from the law of $X_i(t-1)$, as in the mean field approximation, equation 3.7 becomes

$$\mathbb{P}(X_i(t) = 1 \mid X_i(t-1) = 0) = h_{t-1}(1) \tag{3.8}$$

where $h_{t-1} = \text{Law}(X_i(t-1))$ and so $h_{t-1}(1) = \mathbb{P}(X_i(t-1) = 1)$. Consequenlty,

$$\tilde{Q}(1, 0, h_{t-1}) = h_{t-1}(1). \tag{3.9}$$

3.1.2 Implementation

We now discuss an implementation of the mean field approximation for discrete time interacting particle systems.

Given that $X_i(0) \sim \nu$ we may use $\tilde{Q}$ to obtain $h_T = \text{Law}(X_i(T))$ under the assumptions of the mean field approximation. This dynamic programming algorithm is given below.

Algorithm 3 MeanFieldApproximation($\tilde{Q}, \nu, T$)

```
$h_0 = \nu$
for $t = 1, 2, ..., T$ do
  for $v \in \mathcal{X}$ do
    $h_t(v) = \sum_{u \in \mathcal{X}} \tilde{Q}(v, u, h_{t-1}) h_{t-1}(u)$
  end for
end for
return $h_T$
```

We now aim to prove that h_T is equal to $\text{Law}(X_i(T))$ under the assumptions of the mean field approximation.

Lemma 3.1 $h_T = \text{Law}(X_i(T))$ under the assumption of equation 3.5.

Proof. We may prove our claim by induction. When $T = 0$ we have that $h_0 = \nu$ and $\nu = \text{Law}(X_i(0))$ as needed. Now assume that $h_t = \text{Law}(X_i(t))$ for some $t \geq 0$ and observe that

$$h_{t+1}(v) = \sum_{u \in \mathcal{X}} \tilde{Q}(v, u, h_t) h_t(u) \tag{3.10}$$

$$= \sum_{u \in \mathcal{X}} \mathbb{P}(X_i(t+1) = v \mid X_i(t) = u) \mathbb{P}(X_i(t) = u) \tag{3.11}$$

$$= \sum_{u \in \mathcal{X}} \mathbb{P}(X_i(t+1) = v,\ X_i(t) = u) \tag{3.12}$$

$$= \mathbb{P}(X_i(t+1) = v) \tag{3.13}$$

and therefore $h_{t+1} = \text{Law}(X_i(t+1))$. □

We now showcase some results of algorithm 3. We first consider the contact process on a complete graph as n (the number of particles) grows large. As illustrated by figure 12, the mean field approximation is more accurate with larger n. The simulations are run with the parameters $\mathbb{P}(X_i(0) = 0) = 0.9$, $p = 0.9$, and $q = 0.05$. We plot $\mathbb{P}(X_i(t) = 0)$ for different t, using 10^4 samples to estimate the probability.

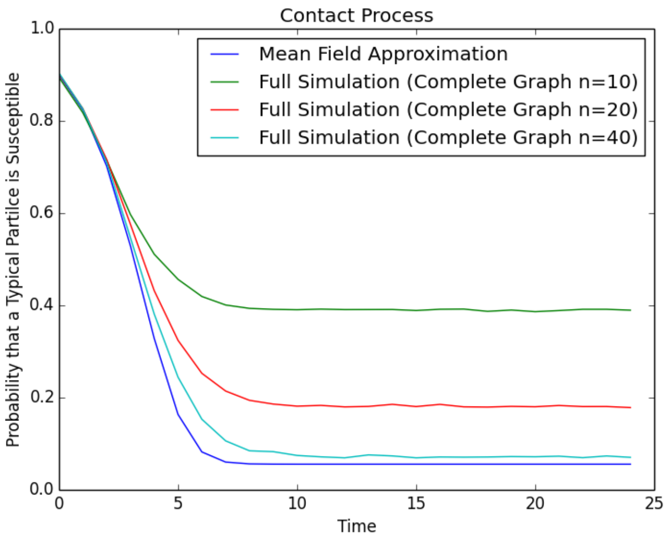

Figure 12: Mean Field Approximation and Full Simulation for the Contact Process

It is important to note that in generating figure 12 we chose parameters which accentuate the discrepancy between different values of n. By changing these parameters slightly the difference is less notable. We generate figure 13 by modifying the simulation in 12 so that the initial condition is $\mathbb{P}(X_i(0) = 0) = 0.5$.

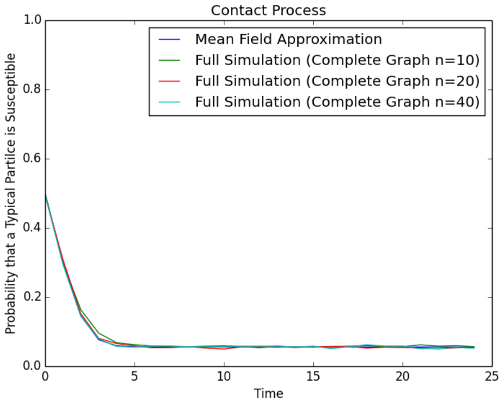

Figure 13: Mean Field Approximation and Full Simulation for the Contact Process with Different Initial Conditions

We now consider the contact process on d-Regular tree of fixed depth. A d-regular tree is a tree graph where each non-leaf particle as d neighbors. As illustrated by the figure below, the mean field approximation grows more accurate with larger d. The simulations are run with the parameters $\mathbb{P}(X_i(0) = 0) = 0.9$, $p = 0.9$, and $q = 0.05$. We plot $\mathbb{P}(X_i(t) = 0)$ for different t, using 10^4 samples to estimate the probability.

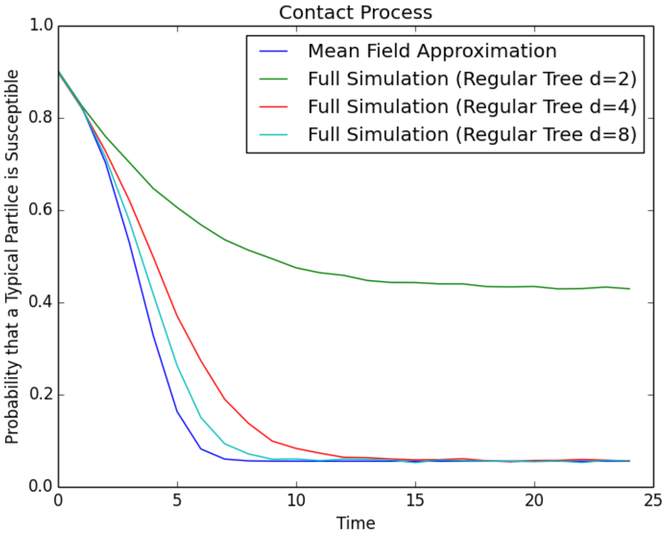

Figure 14: Mean Field Approximation and Full Simulation for the Contact Process with Different Initial Conditions

In summary, computing approximate marginal distributions is computationally efficient and, in many cases, highly accurate with the mean field approximation. However, for sparse graphs the mean field approximation may be very inaccurate.

3.2 Cavity Methods

Various forms of the cavity method have emerged independently throughout the past century. Most notably, in physics the cavity method dates back to 1935 when it was used by Bethe to compute approximate marginals for the Ising model on a lattice [3]. In the 1980s it was rediscovered by Pearl as a procedure for inference in acyclic Bayesian networks [18]. More recently it was used in [13] and [2] to study dynamic systems. The cavity method may also be referred to as belief propagation or the sum-product algorithm.

3.2.1 Cavity Method for Graphical Models

We begin with an introduction of the cavity method in its simplest form. Namely, we consider the problem of performing statistical inference of graphical models. We follow [20, Section 2.4.1] and show that we may compute exact marginals for every node using the cavity method for a Markov random field which respects a finite tree.

Consider a Markov random field $\mathbb{P}$ which respects a tree graph $\mathcal{G}$ with n nodes. By the Hammersly-Clifford theorem of [8] we may factor

$$\mu(x_1, ..., x_n) = \mathbb{P}(X_1 = x_1, ..., X_n = x_n) \tag{3.14}$$

as

$$\mu(x_1, ..., x_n) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(x_i, x_j) \tag{3.15}$$

for some Z, $\{\psi_i\}_{i \in \mathcal{V}}$, $\{\psi_{i,j}\}_{(i,j) \in \mathcal{E}}$.

In the cavity method we wish to compute marginal distribution $\mu_i(x_i) = \mathbb{P}(X_i = x_i)$ for $i \in \mathcal{V}$. To do so we introduce messages which are passed along each edge. The message from node j to i, denoted as $m_{j \to i}(x_i)$, is the belief of j that i is in state x_i. We may form an iterative algorithm to obtain $m_{j \to i}(x_i)$ as follows.

1. Initialize $m^0_{j \to i}(x_i) = m^0_{i \to j}(x_j) = 1$ for each $(i, j) \in \mathcal{E}$, $x_j \in \mathcal{X}$.
2. For $t = 1, 2, ...$ compute $m^t_{j \to i}(x_i)$ by

   $$m^t_{j \to i}(x_i) = \kappa \sum_{x_j} \psi_{i,j}(x_i, x_j) \psi_j(x_j) \prod_{k \in \partial j \setminus \{i\}} m^{t-1}_{k \to j}(x_j) \tag{3.16}$$

   for κ such that $\sum_{x_i} m^t_{j \to i}(x_i) = 1$.
3. Once $m^t_{j \to i}(x_i)$ has converged to $m_{j \to i}(x_i)$, compute the marginal $\mu(x_i)$ as

   $$\mu_i(x_i) = \kappa \psi_i(x_i) \prod_{j \in \partial i} m^t_{j \to i}(x_i). \tag{3.17}$$

   for κ such that $\sum_{x_i} \mu_i(x_i) = 1$. Note that when $m^t_{j \to i}(x_i)$ has converged we omit the superscript t.

This algorithm is often called the *sum-product* algorithm.

Lemma 3.2. The sum-product algorithm for a Markov random field which respects a tree with n nodes converges so that equation 3.17 is correct in at most n iterations.

Proof. We proceed by induction on n. By equation 3.15 a tree with only one node has distribution $\mu(x_1) = \kappa \psi_1(x_1)$. Since there are no edges there are no messages and equation 3.17 is immediately correct. Now assume that the sum-product algorithm on a tree with $n - 1$ nodes converges so that the marginals are correct in at most $n - 1$ iterations.

Consider any tree $\mathcal{G}$ with n nodes $1, ..., n$. Every tree contains at least one leaf and so without loss of generality we label a leaf n. Now let i be the single node with an edge to n. Since n has only one neighbor, the message from n to i converges immediately to

$$m_{n\to i}(x_i) = \kappa_n \sum_{x_n} \psi_{i,n}(x_i, x_n)\psi_n(x_n) \tag{3.18}$$

for some normalization constant κ_n. We now consider nodes $1, ..., n-1$ and observe

$$\mathbb{P}(X_1 = x_1, ..., X_{n-1} = x_{n-1}) \tag{3.19}$$

$$= \mu_{\setminus n}(x_1, ..., x_{n-1}) \tag{3.20}$$

$$= \sum_{x_n} \frac{1}{Z} \prod_{i\in\mathcal{V}} \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \tag{3.21}$$

$$= \frac{1}{\tilde{Z}} \left(\sum_{x_n} \kappa_n \psi_{i,n}(x_i, x_n)\psi_n(x_n) \right) \prod_{i\in\mathcal{V}\setminus\{n\}} \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}\setminus\{(i,n)\}} \psi_{i,j}(x_i, x_j) \tag{3.22}$$

$$= \frac{1}{\tilde{Z}} m_{n\to i}(x_i) \prod_{i\in\mathcal{V}\setminus\{n\}} \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}\setminus\{(i,n)\}} \psi_{i,j}(x_i, x_j) \tag{3.23}$$

for some normalization constant $\tilde{Z}$. When we remove a leaf from a tree we are still left with a tree, and so by assumption the sum-product algorithm will converge to the correct marginals for nodes $1, ..., n-1$ in at most $n-1$ iterations. We are now left to show that after our final iteration we will have correctly computed the marginal of node n. Using the spatial Markov property we may decompose $\mu_n(x_n)$ as

$$\mu_n(x_n) = \sum_{x_i} \mu_{n|i}(x_n|x_i)\mu_1(x_i) \tag{3.24}$$

where $\mu_{n|i}(x_n|x_i) = \mathbb{P}(X_n = x_n \mid X_i = x_i)$ may be further decomposed as follows.

$$\mu_{n|i}(x_n|x_i) = \frac{\mu_{i,n}(x_i, x_n)}{\sum_{x_n} \mu_{i,n}(x_i, x_n)} \tag{3.25}$$

$$= \frac{\psi_n(x_n)\psi_{i,n}(x_i, x_n)}{\sum_{x_n} \psi_n(x_n)\psi_{i,n}(x_i, x_n)} \tag{3.26}$$

$$\propto \frac{\psi_n(x_n)\psi_{i,n}(x_i, x_n)}{m_{n\to i}(x_i)} \tag{3.27}$$

Additionally by our inductive assumption the marginal on node i is correct and so

$$\mu_i(x_i) \propto \psi_i(x_i) m_{n\to i}(x_i) \prod_{k\in\partial i\setminus\{n\}} m_{k\to i}(x_i) \tag{3.28}$$

by equation 3.17. Finally we observe that from equations 3.27 and 3.28 we have

$$\mu_n(x_n) = \sum_{x_i} \mu_{n|i}(x_n|x_i)\mu_1(x_i) \tag{3.29}$$

$$\propto \sum_{x_i} \psi_i(x_i)\psi_n(x_n)\psi_{i,n}(x_i, x_n) \prod_{k\in\partial i\setminus\{n\}} m_{k\to i}(x_i) \tag{3.30}$$

$$\propto \psi_n(x_n)m_{i\to n}(x_n) \tag{3.31}$$

and so equation 3.17 is correct as needed. □

The sum-product algorithm is also used when the graph is no longer a tree. However, when the graph is no longer a tree the marginals may not be exact.

3.2.2 Dynamic Cavity Method

In recent papers by Aurell and Mahmoudi [1, 2] and Lokhov [13, 14] the cavity method has been extended to finite, discrete time, interacting particle systems. Kanoria and Montanari provide a similar method in [9], though it is more specific to majority dynamics. The so-called dynamic cavity method is similar to the sum-product algorithm introduced above. However, the messages are now passed between pairs of particle trajectories. To discuss the dynamic cavity method we first recall the factorization of a finite discrete time interacting particle system given by Lemma 2.3.

$$\mathbb{P}(X^T = x^T) = \prod_{i\in\mathcal{V}} \left(p_i^0(x_i(0)) \prod_{t=1}^{T} w_i(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right) \tag{3.32}$$

Now consider a message $m_{i\to j}(x_i^T \mid x_j^T)$, $i, j \in \mathcal{E}$ defined as the probability that particle i has trajectory x_i^T given the trajectory x_j^t of j in the transformed cavity graph $\mathcal{G}_{\setminus j}$ where node j has beed removed. As emphasized in [9], $m_{i\to j}(x_i^T \mid x_j^T)$ is *not* a conditional probability. Formally the messages satisfy the recursions

$$\begin{aligned} m_{i\to j}(x_i^T \mid x_j^T) = \frac{1}{Z_{ij}} p_i^0(x_i(0)) \sum_{x_{\partial i\setminus j}^T} \left[\prod_{t=1}^{T} w_i(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right] \\ * \prod_{k\in\partial i\setminus j} m_{k\to i}(x_k^T \mid x_i^T). \end{aligned} \tag{3.33}$$

As each message contains an exponential number of components, iterating 3.33 until convergence has exponential computational complexity [14]. Equation 3.33 is analogous to equation (3) in [14] and (6) in [9]. However, (3) contains a small typo concerning the initial conditions.

In Appendix B of [9] the authors show that equation 3.33 is exact in the case where the interaction network is a tree and j is a leaf with neighbor i. Though much of [9] is focused directly on majority dynamics, [14] and [2] proceed by computing marginal distributions. Specifically, in equation (4) of [14] the marginal μ_i for the trajectory of particle i may is given as

$$\begin{aligned}\mu_i(x_i^T) = \frac{1}{Z_i} p_i^0(x_i(0)) \sum_{x_{\partial i}^T} \left[\prod_{t=1}^{T} w_i(x_i(t), x_i(t-1), x_{\partial i}(t-1)) \right] \\ * \prod_{k \in \partial i} m_{k \to i}(x_k^T \mid x_i^T).\end{aligned} \tag{3.34}$$

Lokhov claims that equation 3.34 is exact when $\mathcal{G}$ is a tree. We suggest, however, that even in the case where the graph is a tree equation 3.34 need not exactly hold. This follows from Lemma 2.2 in which we show that the neighbors of particle i are *not* independent in the cavity graph where the trajectory x_i^T is given.

3.2.3 Filling in the Dynamic Cavity Method

We now present a novel extension of the dynamic cavity method which does allow for the exact computation of marginal distributions when the interaction network is acyclic and satisfies the following constraint: There exists a disjoint partition of the particles $\{\mathcal{A}_i\}_{i \in \mathcal{I}}$ where $\partial \mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ implies that $\partial^2 \mathcal{A}_i \cap \mathcal{A}_j = \emptyset$. In other words, if $\mathcal{A}_i$ to $\mathcal{A}_j$ are not connected by an edge then there does not exist a path from $\mathcal{A}_i$ to $\mathcal{A}_j$ of length two. Additionally we let

$$\Delta i = \{ j \ : \ \mathcal{A}_i \text{ and } \mathcal{A}_j \text{ are connected by an edge} \} \tag{3.35}$$

We may always partition a tree in this way and are able to find partitions for some non-tree graphs as well.

We now consider a message $m_{i \to j}(x_{\mathcal{A}_i}^T \mid x_{\mathcal{A}_j}^T)$, for $\mathcal{A}_i, \mathcal{A}_j$ connected by an edge. The message $m_{i \to j}(x_{\mathcal{A}_i}^T \mid x_{\mathcal{A}_j}^T)$ is defined as the probability that particles $\mathcal{A}_i$ have trajectory $x_{\mathcal{A}_i}^T$ given the trajectory $x_{\mathcal{A}_j}^T$ of particles $\mathcal{A}_j$ in the transformed cavity graph $\mathcal{G}_{\setminus \mathcal{A}_j}$ where particles $\mathcal{A}_j$ have beed removed. Formally the messages satisfy the recursions

$$\begin{aligned} m_{i \to j}(x_{\mathcal{A}_i}^T \mid x_{\mathcal{A}_j}^T) = \frac{1}{Z_{ij}} \left(\prod_{u \in \mathcal{A}_i} p_u^0(x_u(0)) \right) \\ * \sum_{\substack{x_{\mathcal{A}_j}^T \\ j \in \Delta i}} \left[\prod_{t=1}^{T} \prod_{u \in \mathcal{A}_i} w_u(x_u(t), x_u(t-1), x_{\partial u}(t-1)) \right] \\ * \prod_{k \in \Delta i \setminus j} m_{k \to i}(x_{\mathcal{A}_k}^T \mid x_{\mathcal{A}_i}^T). \end{aligned} \tag{3.36}$$

and we may then find marginal distributions over the local region $\mathcal{A}_i$ as

$$\begin{aligned}\mu_{\mathcal{A}_i}(x^T_{\mathcal{A}_i}) = \frac{1}{Z_i}&\left(\prod_{u\in\mathcal{A}_i} p^0_u(x_u(0))\right)\\ &* \sum_{\substack{x^T_{\mathcal{A}_j}\\ j\in\Delta i}}\left[\prod_{t=1}^{T}\prod_{u\in\mathcal{A}_i} w_u(x_u(t), x_u(t-1), x_{\partial u}(t-1))\right]\\ &* \prod_{k\in\Delta i} m_{k\to i}(x^T_{\mathcal{A}_k} \mid x^T_{\mathcal{A}_i}).\end{aligned} \tag{3.37}$$

From $\mu_{\mathcal{A}_i}(x^T_{\mathcal{A}_i})$ we may further marginalize to obtain marginals for single particle trajectories. Since the interaction network is acyclic, for any $i, j \in \Delta k$ we know from Lemma 2.5 that the trajectories of $\mathcal{A}_i$ and $\mathcal{A}_j$ are independent in the cavity graph where the trajectory for $\mathcal{A}_k$ is given.

We conclude by noting that in general the dynamic cavity method may only be used for small graphs. Unlike our method in 4, the dynamic cavity method requires the marginal to be calculated throughout the graph. Our method allows us to compute marginals for a given local region. The computational complexity of the algorithm we present in 4 does not depend on the number of particles in the overall graph, and may even be applied when the number of particles is infinite.

3.3 Moment Approximations

The pair-approximation and other moment-closure methods are frequently used in biology to increase the tractability of large interacting systems [6, 21]. Though inexact, moment-closure methods may often be a good approximation.

3.3.1 Notation

We first present some notation which may be found in [6]. Given a graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ define $W_i(t)$, $Y_{i,j}(t)$ and $Z_{\ell,i,j}(t)$ as follows.

- $W_i(t)$ is the proportion of particles in state i at time t.
- $Y_{i,j}(t)$ is the proportion of connected particles in states (i, j) at time t.
- $Z_{\ell,i,j}(t)$ is the proportion of connected triplets in states (ℓ, i, j) at time t.

When $\mathcal{G}_s$ is the complete graph then

$$Y_{i,j}(t) = W_i(t)W_j(t) \tag{3.38}$$

as every two particles are connected by an edge.

Moreover, for any graph $\mathcal{G}_s$, the pairwise approximation is given by

$$Z_{\ell,i,j}(t) = \frac{Y_{i,\ell}(t)Y_{i,j}(t)}{W_i(t)}. \tag{3.39}$$

3.3.2 The Power of Two Choices on Graphs: The Mean Field Approximation

We now return to the setting of load balancing on graphs, as discussed in [16, 6]. To build intuition for the pair-approximation we first present the mean field ODE. We restate the model of load balancing as follows. Consider a graph of servers $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ where each server processes jobs in its queue with rate 1. Jobs arrive at each server with a poisson rate λ and for each job arrival at server u_1, u_2 is chosen uniformly from ∂u_1. The job is then allocated to the server among u_1, u_2 with the shortest queue (ties are broken at random).

We now consider the case where $\mathcal{G}_s$ is complete. We follow [6, Section 2.1] in deriving an ODE. Let $P_i(t)$ be the probability that a job arriving at a server with queue length i will be allocated to that server.

$$P_i(t) = \underbrace{\frac{W_i(t)}{2}}_{\text{Neighbor chosen has } i \text{ jobs as well.}} + \underbrace{\sum_{j=i+1}^{\infty} W_j(t)}_{\text{Neighbor chosen has } j > i \text{ jobs.}} \tag{3.40}$$

Using P_i we may obtain an infinite system of ODEs.

$$\begin{aligned} \frac{dW_i}{dt} = & \underbrace{W_{i+1}}_{\text{Exit from queue of length } i+1.} - \underbrace{W_i \mathbb{1}_{\{i>0\}}}_{\text{Exit from queue of length } i.} \\ & + 2\lambda \left(\underbrace{\mathbb{1}_{\{i>0\}} P_{i-1} W_{i-1}}_{\text{Entry to queue length } i-1.} - \underbrace{P_i W_i}_{\text{Entry to queue length } i.} \right) \end{aligned} \tag{3.41}$$

The system converges exactly to the ODE as the number of particles tends to infinity, which follows from Kurtz's theorem [16]. Formally we have that

$$W_i(t) = \mathbb{P}(X_u(t) = i) \tag{3.42}$$

where u is a typical server. In other words, the empirical law of the system converges to the law of a typical server.

3.3.3 The Power of Two Choices on Graphs: the Pair-Approximation is Accurate [6, Section 2.2]

We now follow [6, Section 2.2] in deriving an ODE for a graph where each node has degree k. We denote $Q_i(t)$ as the probability that a job that arrives at a server with i jobs is allocated to that server.

$$Q_i(t) = \frac{\frac{Y_{i,i}(t)}{2} + \sum_{j=i+1}^{\infty} Y_{i,j}(t)}{W_i(t)} \tag{3.43}$$

We divide by $W_i(t)$ as we are given that one of the servers in the pair has i jobs. We may consider instead the equivalent formulation where jobs are incident on each edge (u_1, u_2) with rate λ and allocated to the server with the fewest job (ties broken at random).

There are three ways in which $Y_{i,j}$ can evolve.

1. **A job is served.** We add to $\frac{dY_{i,j}}{dt}$ the term

$$\underbrace{Y_{i+1,j} + Y_{i,j+1}}_{\text{Exit from a queue of length } i+1 \text{ or } j+1.} \quad - \quad \underbrace{Y_{i,j}\mathbb{1}_{\{i>0\}} - Y_{i,j}\mathbb{1}_{\{j>0\}}}_{\text{Exit from one queue of pair with jobs } i,j.} \tag{3.44}$$

2. **A job arrives at a pair of servers with queue lengths $(i-1, j)$, $(i, j-1)$, or (i, j).** There are $nk/2$ edges and so a job arrives at each edge with rate $2\lambda/k$. We add to $\frac{dY_{i,j}}{dt}$ the term

$$\frac{2\lambda}{k}\left(Y_{i-1,j}a(i-1,j) + Y_{i,j-1}a(j-1,i) - Y_{i,j}\right) \tag{3.45}$$

   where $a(x, y) = 1$ is $x < y$, $1/2$ if $x = y$, and 0 if $x > y$.

3. **A job arrives at an edge incident to a server with i or j jobs.** For a pair of servers with queue length (i, j) a job will enter at a different edge and be allocated to i with probability $R_{ij}(t) = \left(Z_{i,i,j}(t)/2 + \sum_{\ell=i+1}^{\infty} Z_{\ell,i,j}(t)\right)/Y_{ij}(t)$. Using the pair-approximation $R_{i,j}(t) = Q_i(t)$ and so we add the following term to $\frac{dY_{i,j}}{dt}$.

$$\frac{2\lambda(k-1)}{k}\left(Q_{i-1}Y_{i-1,j}\mathbb{1}_{\{i>0\}} + Q_{j-1}Y_{i,j-1}\mathbb{1}_{\{j>0\}} - (Q_i + Q_j)Y_{i,j}\right) \tag{3.46}$$

   The factor of $k - 1$ arises each server has k neighbors.

$\frac{dY_{i,j}}{dt}$ is then the addition of terms 3.44, 3.45 and 3.46. This equation is not exact and has no known close form. However, Gast simulates the ODE to obtain very accurate results.

4 Local Recursions

We now present our novel local approximations for the dynamics for a typical particle building on the conditional independence result of Lacker, Ramanan, and Wu in [11].

In this section we will be considering a discrete time interacting particle system $X = \{X_i(t)\}_{t\in\mathbb{N}, i\in\mathcal{V}}$ with interaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For certain local regions of particles $\mathcal{A} \subseteq \mathcal{V}$ our algorithm will compute the joint distribution of $X_{\mathcal{A}}^T = \{X_i(t)\}_{i\in\mathcal{A}, t\leq T}$. We denote the joint distribution as $J_{\mathcal{A}}^T$ where

$$J_{\mathcal{A}}^T(x_{\mathcal{A}}^T) = \mathbb{P}\left(X_{\mathcal{A}}^T = x_{\mathcal{A}}^T\right). \tag{4.1}$$

In general, we refer to $X_{\mathcal{A}}^T$ as the *full trajectory* of $\mathcal{A}$. From the joint distribution we may marginalize to obtain the distribution of a typical particle trajectory. The joint distribution of a particle or region of particles is sufficient to fully characterize particle dynamics. And so as N grows large, perhaps infinite, under certain assumptions we will still be able to fully characterize the dynamics of a typical particle. We cannot rely on computations of the full system as N grows large, as they become computationally infeasible.

This section is organized as follows. First we discuss our local recursions when $\mathcal{G}$ is a ring graph. For a ring graph we first present a simpler form of our algorithm which allows us to sample from $J_{\mathcal{A}}^T$. Next we provide our local recursions for the ring graph which will allow us to compute the full joint distribution.

We then present and prove the local recursions for a more general class of graphs, and motivate a fast approximation we call the τ-approximation.

4.1 Ring

Consider the case where the interaction network is a ring graph of N particles.

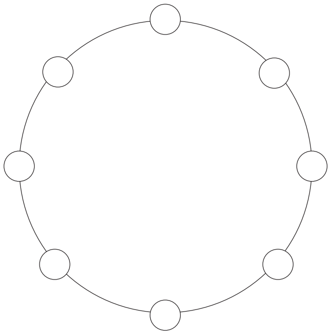

Figure 15: A Ring Graph with 8 Particles

For the ring graph we aim to characterize the dynamics of three consecutive particles. Without loss of generality we label these particles $\mathcal{A} = \{-1, 0, 1\}$, as illustrated by figure 16.

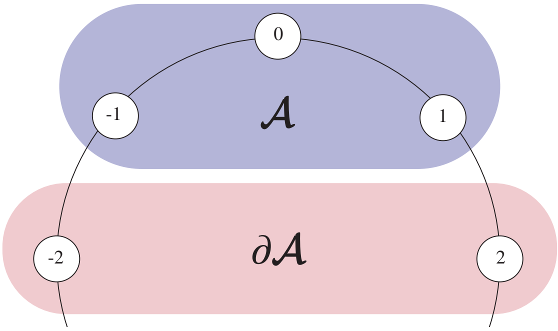

Figure 16: $\mathcal{A} = \{-1, 0, 1\}$

The choice to consider three consecutive particles will soon become clear.

4.1.1 The Local Sampling Algorithm on a Ring

We first provide an algorithm to sample from $J^T_{\mathcal{A}}$. We present this algorithm by induction on time T. Since the initial conditions $\{X_i(0)\}_{i\in\mathcal{V}}$ are i.i.d. we may easily sample $J^0_{\mathcal{A}}$. Assuming that we may sample from $J^t_{\mathcal{A}}$ for $t \geq 0$ we will describe the process for sampling $x^{t+1}_{\mathcal{A}} = (x_{\mathcal{A}}(0), ..., x_{\mathcal{A}}(t+1))$ from $J^{t+1}_{\mathcal{A}}$.

We account for the value of $x_{\mathcal{A}}$ at times $0, 1, ..., t$ by invoking our assumption to sample $x^t_{\mathcal{A}} = (x_{\mathcal{A}}(0), ..., x_{\mathcal{A}}(t))$ from $J^t_{\mathcal{A}}$. We are now left only to sample for the final time step.

Since a particle updates according to its own state and the state of its neighbors, we may easily obtain a sample for the final time step if we can sample the state of particles $\partial\mathcal{A} = \{-2, 2\}$ at time t. To obtain samples for the state of particles $\{-2, 2\}$ at time t we use the property of conditional independence given the double boundary.

As the problem is symmetric we only describe in detail the process of sampling the state of particle -2 at time t. Formally, we need to sample state of particle -2 at time t conditioned on the state of particles $\{-1, 0, 1\}$ at times $0, 1, ..., t$. By the property of conditional independence given the double boundary (specifically Lemma 2.5) we may ignore particle 1 as it is outside the double boundary of -2. Instead we need only sample from the state of particle -2 at time t conditioned on the state of particles -1 and 0 at times $0, 1, ..., t$. We use now that all particles are identical, and so the state of particle -2 conditioned on the state of particles -1 and 0 at times

$0, 1..., t$ is equal in distribution to the state of particle -1 conditioned on the state of particles 0 and 1 at times $0, 1, ..., t$. So we may repeatedly sample $y^t_{\mathcal{A}}$ from $J^t_{\mathcal{A}}$ until we find that $x^t_{-1} = y^t_0$ and $x^t_0 = y^t_1$, in which case we let $y_{-1}(t)$ be our sample for particle -2 at time t.

In algorithm 4 we provide pseudocode for the local sampling process we have described above. The algorithm returns a sample from $J^T_{\mathcal{A}}$ given the following arguments.

- Time T up to which the algorithm should be run.
- Update function f where $X_i(t+1) = f(X_i(t), X_{\partial i}(t), \xi_i(t+1))$.
- Initial conditions ν where $\{x_i(0)\}$ i.i.d. ν. It is assumed we can sample from ν in constant time.
- Independent random variables $\xi = \{\xi_i(t)\}_{s \leq t}$. It is assumed we can sample from $\text{Law}(\xi_i(t))$ in constant time.

Algorithm 4 LocalRingSample(T, f, ν, ξ)

```
Sample $x_{-1}(0), x_0(0), x_1(0)$ from $\nu$.

for $t = 0, 1, ..., T-1$ do

    {Update the middle particle}
    Sample $r_0(t+1)$ from $(\xi_0(t+1))$
    $x_0(t+1) = f\left(x_0(t), \left[x_{-1}(t), x_1(t)\right], r_0(t+1)\right)$

    {Update the left particle}
    while $\{y_0(s), y_1(s)\}_{s \leq t} \neq \{x_{-1}(s), x_0(s)\}_{s \leq t}$ do
        $\{y_{-1}, y_0(s), y_1(s)\}$ = LocalRingSample($t, f, \nu, \xi$)
    end while
    Sample $r_{-1}(t+1)$ from Law$(\xi_{-1}(t+1))$
    $x_{-1}(t+1) = f\left(x_{-1}(t), \left[y_{-1}(t), x_0(t)\right], r_{-1}(t+1)\right)$

    {Update the right particle}
    while $\{y_{-1}(s), y_0(s)\}_{s \leq t} \neq \{x_0(s), x_1(s)\}_{s \leq t}$ do
        $\{y_{-1}, y_0(s), y_1(s)\}$ = LocalRingSample($t, f, \nu, \xi$)
    end while
    Sample $r_1(t+1)$ from Law$(\xi_1(t+1))$
    $x_1(t+1) = f\left(x_1(t), \left[x_0(t), y_1(t)\right], r_1(t+1)\right)$

end for
return $\{x_{-1}(s), x_0(s), x_1(s)\}_{s \leq T}$
```

4.1.2 Local Recursions on a Ring

We recognize two main flaws in algorithm 4. Since we may only sample from $J_{\mathcal{A}}^T$ we need to run the algorithm many times to understand particle dynamics. Additionally, the algorithm is not deterministic and so the number of recursive calls we make may vary. We aim to remedy the shortcomings of algorithm 4 by presenting deterministic recursions for the full joint distribution $J_{\mathcal{A}}^T$. This section is meant to build intuition for the more general local recursions which we present formally and prove in Section 4.2.

Instead of an update function f we now consider g where

$$\mathbb{P}(X_i(t) = v \mid X_i(t-1) = u) = g(v, u, X_{\partial i}(t-1)). \tag{4.2}$$

As we discuss in Remark 2.1 these two formulations are equivalent.

To explain our local recursions it is first necessary to introduce c_ℓ^t and c_r^t, given by

$$\begin{aligned}&c_\ell^t\left(x_{-1}(t),\{x_0(t),x_1(t)\}_{s\leq t}\right)\\&\quad=\mathbb{P}\left(X_{-1}(t)=x_{-1}(t)\mid\{X_0(t),X_1(t)\}_{s\leq t}=\{x_0(t),x_1(t)\}_{s\leq t}\right)\end{aligned}\tag{4.3}$$

and

$$\begin{aligned}&c_r^t\left(x_1(t),\{x_{-1}(t),x_0(t)\}_{s\leq t}\right)\\&\quad=\mathbb{P}\left(X_1(t)=x_1(t)\mid\{X_{-1}(t),X_0(t)\}_{s\leq t}=\{x_{-1}(t),x_0(t)\}_{s\leq t}\right)\end{aligned}\tag{4.4}$$

defined only when the event on which we condition has positive probability.

We may think of c_ℓ^t and c_r^t as the left and right conditional distributions respectively. The probability that particle -1 is in a given state at time t conditioned on the state of particles 0 and 1 at times $0,1,...,t$ is given by c_ℓ^t. Likewise c_r^t gives the probability that particle 1 is in a given state at time t conditioned on the state of particles -1 and 0 at times $0,1,...,t$.

We now proceed by induction on $J_{\mathcal{A}}^t$, c_ℓ^t and c_r^t, as illustrated by figure 17. We are given $J_{\mathcal{A}}^0$ by the initial conditions. We then show that

1. given $J_{\mathcal{A}}^t$ we may compute c_ℓ^t and c_r^t and
2. given $J_{\mathcal{A}}^t$, c_ℓ^t and c_r^t we may compute $J_{\mathcal{A}}^{t+1}$.

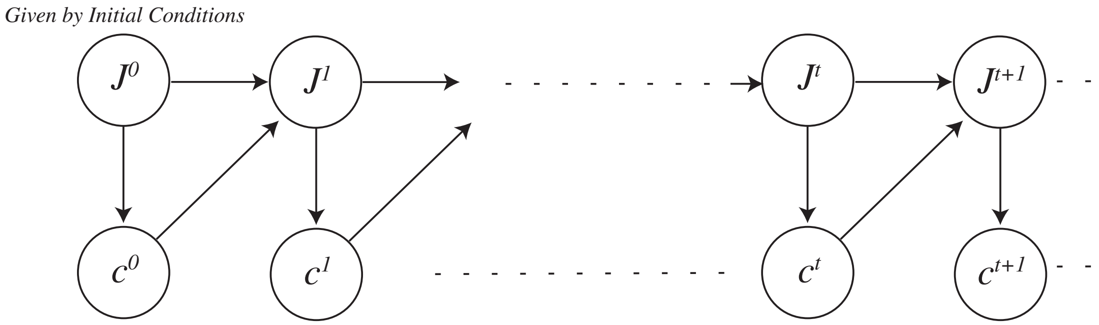

Figure 17: Induction on J^t and c^t.

It is immediate that we may compute c_ℓ^t and c_r^t from $J_{\mathcal{A}}^t$. For example we have

$$\begin{aligned}&c_\ell^t\left(x_{-1}(t),\{x_0(t),x_1(t)\}_{s\leq t}\right)\\&\quad=\frac{\sum_{\{y_{-1}(s)\}_{s<t},\ y_{-1}(t)=x_{-1}(t)}J_{\mathcal{A}}^t\left(\{y_{-1}(t),x_0(t),x_1(t)\}_{s\leq t}\right)}{\sum_{\{y_{-1}(s)\}_{s\leq t}}J_{\mathcal{A}}^t\left(\{y_{-1}(t),x_0(t),x_1(t)\}_{s\leq t}\right)}\end{aligned}\tag{4.5}$$

for c^t_ℓ. And so the challenge that remains is to compute $J^{t+1}_{\mathcal{A}}$ from $J^t_{\mathcal{A}}, c^t_\ell, c^t_r$. We observe that

$$J^{t+1}_{\mathcal{A}}\left(x^{t+1}_{\mathcal{A}}\right) = \mathbb{P}\left(X^{t+1}_{\mathcal{A}} = x^{t+1}_{\mathcal{A}}\right) \tag{4.6}$$

$$= \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) \mathbb{P}\left(X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right). \tag{4.7}$$

$\mathbb{P}\left(X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right)$ is simply $J^t_{\mathcal{A}}\left(x^t_{\mathcal{A}}\right)$ which we are assuming that we have. Therefore, we need only calculate $\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right)$. We notice that

$$\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) \tag{4.8}$$

$$\begin{aligned} = \sum_{y_{-2}, y_2} & \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}},\ X_{-2}(t) = y_{-2},\ X_2(t) = y_2\right) \\ & * \mathbb{P}\left(X_{-2}(t) = y_{-2},\ X_2(t) = y_2 \;\middle|\; \vec{X}^t_{\mathcal{A}} = \vec{x}^t_{\mathcal{A}}\right) \end{aligned} \tag{4.9}$$

and now examine the two terms in the summation above separately. We simplify the first term as

$$\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}},\ X_{-2}(t) = y_{-2},\ X_2(t) = y_2\right) \tag{4.10}$$

$$\begin{aligned} &= g\left(x_{-1}(t+1), x_{-1}(t), \left[y_{-2}, x_0(t)\right]\right) \\ &\quad * g\left(x_0(t+1), x_0(t), \left[x_{-1}(t), x_1(t)\right]\right) \\ &\quad * g\left(x_1(t+1), x_1(t), \left[x_0(t), y_2\right]\right) \end{aligned} \tag{4.11}$$

Finally we use conditional independence given the double boundary (specifically Lemma 2.5) to simplify the latter term.

$$\mathbb{P}\left(X_{-2}(t) = y_{-2},\ X_2(t) = y_2 \;\middle|\; X^T_{\mathcal{A}} = x^T_{\mathcal{A}}\right) \tag{4.12}$$

$$= \mathbb{P}\left(X_{-2}(t) = y_{-2} \;\middle|\; X^T_{\mathcal{A}} = x^T_{\mathcal{A}}\right) * \mathbb{P}\left(X_2(t) = y_2 \;\middle|\; X^T_{\mathcal{A}} = x^T_{\mathcal{A}}\right) \tag{4.13}$$

$$\begin{aligned} &= \mathbb{P}\left(X_{-2}(t) = y_{-2} \;\middle|\; X^T_{-1} = x^T_{-1},\ X^T_0 = x^T_0\right) \\ &\quad * \mathbb{P}\left(X_2(t) = y_2 \;\middle|\; X^T_0 = x^T_0,\ X^T_1 = x^T_1\right) \end{aligned} \tag{4.14}$$

$$= c^t_\ell\left(y_{-2}, \{x_{-1}(t), x_0(t)\}_{s \le t}\right) * c^t_r\left(y_2, \{x_0(t), x_1(t)\}_{s \le t}\right) \tag{4.15}$$

So in summary we have

$$\begin{aligned} & J^{t+1}_{\mathcal{A}}\left(x^{t+1}_{\mathcal{A}}\right) \\ &= J^t_{\mathcal{A}}\left(x^t_{\mathcal{A}}\right) \sum_{y_{-2}, y_2} c^t_\ell\left(y_{-2}, \{x_{-1}(t), x_0(t)\}_{s \le t}\right) * c^t_r\left(y_2, \{x_0(t), x_1(t)\}_{s \le t}\right) \\ &\qquad\qquad * g\left(x_{-1}(t+1), x_{-1}(t), \left[y_{-2}, x_0(t)\right]\right) \\ &\qquad\qquad * g\left(x_0(t+1), x_0(t), \left[x_{-1}(t), x_1(t)\right]\right) \\ &\qquad\qquad * g\left(x_1(t+1), x_1(t), \left[x_0(t), y_2\right]\right) \end{aligned} \tag{4.16}$$

and we may indeed compute $J_{\mathcal{A}}^{t+1}$ from $J_{\mathcal{A}}^{t}, c_{\ell}^{t}$ and c_{r}^{t}.

Finally we note that to avoid repeated computations we may factor equation 4.16 as

$$\begin{aligned}
&J_{\mathcal{A}}^{t+1}\left(x_{\mathcal{A}}^{t+1}\right) \\
&= J_{\mathcal{A}}^{t}\left(x_{\mathcal{A}}^{t}\right) g\left(x_0(t+1), x_0(t), \left[x_{-1}(t), x_1(t)\right]\right) \\
&* \left(\sum_{y_{-2}} g\left(x_{-1}(t+1), x_{-1}(t), \left[y_{-2}, x_0(t)\right]\right) * c_{\ell}^{t}\left(y_{-2}, \{x_{-1}(t), x_0(t)\}_{s\leq t}\right)\right) \qquad (4.17)\\
&* \left(\sum_{y_{2}} g\left(x_{1}(t+1), x_{1}(t), \left[x_0(t), y_2\right]\right) * c_{r}^{t}\left(y_2, \{x_0(t), x_1(t)\}_{s\leq t}\right)\right).
\end{aligned}$$

4.2 Local Recursions for More General Graphs

We now present our local recursions for discrete time interacting particle systems on a more general class of graphs. We consider a discrete time interacting particle system with interaction network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We give the joint distribution for a local region $\mathcal{A} \subseteq \mathcal{V}$ when the following properties hold.

(A1) All particles are identical. And so for any two isomorphic subgraphs $G_{\mathcal{I}}$ and $G_{\mathcal{H}}$ we have that

$$X_{\mathcal{I}}^{T} \overset{d}{=} X_{\mathcal{H}}^{T}. \qquad (4.18)$$

for all T.

Note that for any set $\mathcal{I} \subseteq \mathcal{V}$ we let $\mathcal{G}_{\mathcal{I}}$ denote the subgraph consisting of particles in the set $\mathcal{I}$. We say that two subgraphs $\mathcal{G}_{\mathcal{I}}$ and $\mathcal{G}_{\mathcal{H}}$ are isomorphic ($\mathcal{G}_{\mathcal{I}} \simeq \mathcal{G}_{\mathcal{H}}$) if there exists a bijection $\phi : \mathcal{I} \to \mathcal{H}$ such that

$$(u, v) \in \mathcal{G}_{\mathcal{I}} \iff (\phi(u), \phi(v)) \in \mathcal{G}_{\mathcal{H}}. \qquad (4.19)$$

(A2) There exists a disjoint partition $\mathcal{A}_1, ..., \mathcal{A}_k$ of $\mathcal{A}$, and an associated disjoint parition $\mathcal{B}_1, ..., \mathcal{B}_k$ of $\mathcal{B} = \partial\mathcal{A}$ (where $\mathcal{B}_i$ may be empty) satisfying $\partial\mathcal{A}_i \cap \mathcal{B}_j = \emptyset$ for each $i \neq j$.

And so particles in $\mathcal{A}_i$ update only according to the state of particles in $\mathcal{B}_i \cup \mathcal{A}$.

(A3) For each non-empty $\mathcal{B}_i$ and $\mathcal{B}_j$ with $i \neq j$, every path

$$\{(u_0, u_1), (u_1, u_2), ..., (u_{k-1}, u_k)\} \subseteq \mathcal{E}$$

with $u_0 \in \mathcal{B}_i, u_k \in \mathcal{B}_j$ contains at least two particles $u_j \in \mathcal{A}$.

And so by Lemma 2.5 we have that $X_{\mathcal{B}_i}^{T} \perp\!\!\!\perp X_{\mathcal{B}_j}^{T} \mid X_{\mathcal{A}}^{T}$.

(A4) For each $\mathcal{B}_i \neq \emptyset$ we let $\mathcal{D}_i$ denote the overlap between the double boundary of $\mathcal{B}_i$ and $\mathcal{A}$. Formally, $\mathcal{D}_i = \partial^2 \mathcal{B}_i \cap \mathcal{A}$. For each $\mathcal{B}_i \neq \emptyset$ there must exist $\mathcal{C}_i, \mathcal{R}_i \subseteq \mathcal{A}$ such that $G_{\mathcal{R}_i} \simeq G_{\mathcal{B}_i}$, $G_{\mathcal{C}_i} \simeq G_{\mathcal{D}_i}$, and $G_{\mathcal{R}_i \cup \mathcal{C}_i} \simeq G_{\mathcal{B}_i \cup \mathcal{D}_i}$.

In our recursions we will be using $\mathcal{R}_i$ as a substitute for $\mathcal{B}_i$.

Once again we denote the joint distribution for the local region $\mathcal{A}$ as

$$J_{\mathcal{A}}^T(x_A^T) = \mathbb{P}\left(X_A^T = x_A^T\right). \tag{4.20}$$

We now have a family of conditional distributions c_i^T for $i = 1, ..., k$ where

$$c_i^T\left(x_{\mathcal{B}_i}(T), x_{\mathcal{D}_i}^T\right) = \mathbb{P}\left(X_{\mathcal{B}_i}(T) = x_{\mathcal{B}_i}(T) \;\middle|\; X_{\mathcal{D}_i}^T = x_{\mathcal{D}_i}^T\right). \tag{4.21}$$

defined only when $\mathbb{P}(X_{\mathcal{D}_i}^T = x_{\mathcal{D}_i}^T) > 0$.

Lemma 4.1. Under the assumptions (A1) through (A4) we may compute $J_{\mathcal{A}}^T$ for any $T \in \mathbb{N}$.

Proof. We proceed by induction on $J_{\mathcal{A}}^t$, c_ℓ^t and c_r^t, as illustrated by figure 18. We are given $J_{\mathcal{A}}^0$ by the initial conditions. By Lemmas 3.2 and 3.3 we show that

1. given $J_{\mathcal{A}}^t$ we may compute $\{c_i^t\}_{i=1,..,k}$ (Lemma 4.2) and
2. given $J_{\mathcal{A}}^t$, $\{c_i^t\}_{i=1,..,k}$ we may compute $J_{\mathcal{A}}^{t+1}$ (Lemma 4.3).

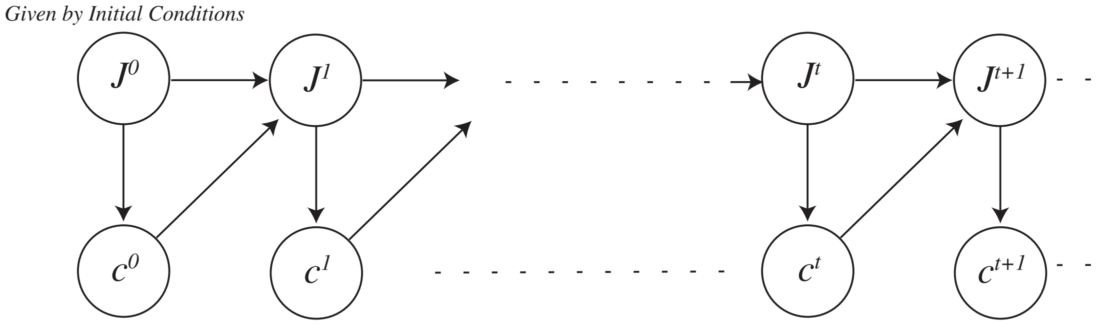

Figure 18: Induction on J^t and c^t.

Lemma 4.2. Under the assumptions (A1) through (A4) we may compute $\{c_i^t\}_{i=1,..,k}$ given $J_{\mathcal{A}}^t$.

Proof. Consider any $i \in \{1, ..., k\}$. By assumptions (A1) and (A4) we have that

$$X_{\mathcal{B}_i \cup \mathcal{D}_i}^t \overset{d}{=} X_{\mathcal{R}_i \cup \mathcal{C}_i}^t \tag{4.22}$$

for $\mathcal{R}_i, \mathcal{C}_i \subseteq \mathcal{A}$. We also assume that we are given $J^t_{\mathcal{A}}$, the joint distribution of $X^t_{\mathcal{A}}$. And so for $x^t_{\mathcal{B}_i}$, $x^t_{\mathcal{D}_i}$ with $\mathbb{P}\left(X^t_{\mathcal{D}_i} = x^t_{\mathcal{D}_i}\right) > 0$ we observe that

$$\begin{aligned}
&c^t_i\left(x_{\mathcal{B}_i}(t), x^t_{\mathcal{D}_i}\right) && (4.23)\\
&= \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X^t_{\mathcal{D}_i} = x^t_{\mathcal{D}_i}\right) && (4.24)\\
&= \mathbb{P}\left(X_{\mathcal{R}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X^t_{\mathcal{C}_i} = x^t_{\mathcal{D}_i}\right) \qquad \text{by equation 4.22} && (4.25)\\
&= \frac{\sum_{y^t_{\mathcal{A}}\ :\ y_{\mathcal{R}_i}(t) = x_{\mathcal{B}_i}(t),\ y^t_{\mathcal{C}_i} = x^t_{\mathcal{D}_i}} J^t_{\mathcal{A}}\left(y^t_{\mathcal{A}}\right)}{\sum_{y^t_{\mathcal{A}}\ :\ y^t_{\mathcal{C}_i} = x^t_{\mathcal{D}_i}} J^t_{\mathcal{A}}\left(y^t_{\mathcal{A}}\right)} && (4.26)
\end{aligned}$$

and we may compute c^t_i from $J^t_{\mathcal{A}}$ as needed. □

From equation 4.26 it is immediate that if $\mathcal{R}_i = \mathcal{R}_j$ and $\mathcal{C}_i = \mathcal{C}_j$ then $c^t_i = c^t_j$. In the implementation of our local recursions we use this to avoid redundant computation.

Lemma 4.3. Under the assumptions (A1) through (A4) we may compute $J^{t+1}_{\mathcal{A}}$ given $J^t_{\mathcal{A}}$ and $\{c^t_i\}_{i=1,..,k}$.

Proof.

We observe that

$$\begin{aligned}
J^{t+1}_{\mathcal{A}}\left(x^{t+1}_{\mathcal{A}}\right) &= \mathbb{P}\left(X^{t+1}_{\mathcal{A}} = x^{t+1}_{\mathcal{A}}\right) && (4.27)\\
&= \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) \mathbb{P}\left(X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right). && (4.28)
\end{aligned}$$

As $\mathbb{P}\left(X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right)$ is simply $J^t_{\mathcal{A}}\left(x^t_{\mathcal{A}}\right)$ which we have by assumption, we are now left to compute $\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \middle| X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right)$. We notice that

$$\begin{aligned}
&\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \middle| X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) && (4.29)\\
&= \sum_{x_{\mathcal{B}}(t)} \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}},\ X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t)\right) \\
&\qquad * \mathbb{P}\left(X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) && (4.30)
\end{aligned}$$

and now examine the two terms in the summation above separately. We may simplify the first term as

$$\begin{aligned}
&\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}},\ X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t)\right) \\
&= \prod_{v \in \mathcal{A}} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) && (4.31)
\end{aligned}$$

Finally we use conditional independence given the double boundary (specifically Lemma 2.5) to simplify the latter term.

$$\mathbb{P}\left(X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t) \;\middle|\; X^t_{\mathcal{A}} = x^t_{\mathcal{A}}\right) \qquad (4.32)$$

$$= \prod_{i=1}^{k} \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X_{\mathcal{A}}^t = x_{\mathcal{A}}^t \right) \quad \text{by (A3)} \tag{4.33}$$

$$= \prod_{i=1}^{k} \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X_{\mathcal{D}_i}^t = x_{\mathcal{D}_i}^t \right) \quad \text{by (A4)} \tag{4.34}$$

$$= \prod_{i=1}^{k} c_i^t \left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^t \right) \tag{4.35}$$

In summary we have

$$\begin{aligned} &J_{\mathcal{A}}^{t+1}\left(x_{\mathcal{A}}^{t+1}\right) \\ &= J_{\mathcal{A}}^t\left(x_{\mathcal{A}}^t\right) \sum_{x_{\mathcal{B}}(t)} \prod_{v \in \mathcal{A}} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \prod_{i=1}^{k} c_i^t\left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^t\right) \end{aligned} \tag{4.36}$$

and so we may compute $J_{\mathcal{A}}^{t+1}$ from $J_{\mathcal{A}}^t, c_\ell^t$ and $\{c_i^t\}_{i=1,..,k}$. □

To avoid redundant computation in our implementation we observe that equation 4.36 factors as

$$\begin{aligned} &J_{\mathcal{A}}^{t+1}\left(x_{\mathcal{A}}^{t+1}\right) \\ &= J_{\mathcal{A}}^t\left(x_{\mathcal{A}}^t\right) \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i = \emptyset}} \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \\ &\quad * \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i \neq \emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c_i^t\left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^t\right) \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right). \end{aligned} \tag{4.37}$$

Our local recursions may then be expressed as a dynamic programming algorithm as follows.

Algorithm 5 LocalRecursions(T, ν, g)

```
for each $x_{\mathcal{A}}(0)$ do
  $J^0_{\mathcal{A}}(x_{\mathcal{A}}(0)) = \prod_{v \in \mathcal{A}} \nu(x_v(0))$
end for

for $t = 0, 1, ..., T-1$ do

  {compute each $c^t_i$}
  for $i = 1, 2, ..., k$ do
    if $\mathcal{R}_i = \mathcal{R}_j$ and $\mathcal{C}_i = \mathcal{C}_j$ for some $j < i$ then
      $c^t_i = c^t_j$
    else
      for each $x_{\mathcal{B}_i}(t)$, and $x^t_{\mathcal{D}_i}$ do
        $c^t_i(x_{\mathcal{B}_i}(t), x^t_{\mathcal{D}_i}) = \frac{\sum_{y^t_{\mathcal{A}} \,:\, y_{\mathcal{R}_i}(t)=x_{\mathcal{B}_i}(t),\ y^t_{\mathcal{C}_i}=x^t_{\mathcal{D}_i}} J^t_{\mathcal{A}}(y^t_{\mathcal{A}})}{\sum_{y^t_{\mathcal{A}} \,:\, y^t_{\mathcal{C}_i}=x^t_{\mathcal{D}_i}} J^t_{\mathcal{A}}(y^t_{\mathcal{A}})}$
      end for
    end if
  end for

  {compute $J^{t+1}_{\mathcal{A}}$}
  for each $x^{t+1}_{\mathcal{A}}$ do
    $J^{t+1}_{\mathcal{A}}(x^{t+1}_{\mathcal{A}}) = J^t_{\mathcal{A}}(x^t_{\mathcal{A}}) \prod_{\substack{i \in \{1,...,k\} \\ \mathcal{B}_i = \emptyset}} \prod_{v \in \mathcal{A}_i} g(x_v(t+1), x_v(t), x_{\partial v}(t))$
        $* \prod_{\substack{i \in \{1,...,k\} \\ \mathcal{B}_i \neq \emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c^t_i(x_{\mathcal{B}_i}(t), x^t_{\mathcal{D}_i}) \prod_{v \in \mathcal{A}_i} g(x_v(t+1), x_v(t), x_{\partial v}(t))$
  end for
end for
return $J^T_{\mathcal{A}}$
```

We now present a series of interaction networks on which our local recursions may be applied. For each interaction network we provide regions $\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, \mathcal{R}_i$ which satisfy properties (A2), (A3) and (A4). We may ignore (A1) as it will always follow from our assumption that each particle is identical.

4.2.1 Revisiting The Ring

We may now express our local recursions on a ring once more in terms of the more general formulation.

We still consider $\mathcal{A} = \{-1, 0, 1\}$ with $\partial\mathcal{A} = \mathcal{B} = \{-2, 2\}$. However we now partition $\mathcal{A}$ as $\mathcal{A}_1 = \{-1\}$, $\mathcal{A}_2 = \{-1\}$, $\mathcal{A}_3 = \{0\}$ and $\mathcal{B}$ as $\mathcal{B}_1 = \{-2\}$, $\mathcal{B}_2 = \{2\}$, $\mathcal{B}_3 = \emptyset$. Notice that (A2) is satisfied as $\partial\mathcal{A}_i \cap \mathcal{B}_j = \emptyset$ for each $i \neq j$.

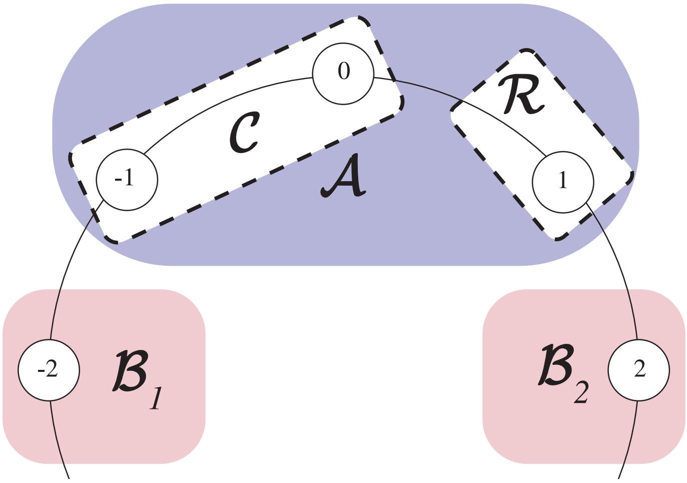

Figure 19: Local Recursions for a Ring

We observe that $\mathcal{D}_1 = \partial^2\mathcal{B}_1 \cap \mathcal{A} = \{-1, 0\}$, $\mathcal{D}_2 = \partial^2\mathcal{B}_2 \cap \mathcal{A} = \{1, 0\}$ and must find $\mathcal{C}_i, \mathcal{R}_i$ which satisfy

$$G_{\mathcal{B}_i} \simeq G_{\mathcal{R}_i},\ G_{\mathcal{D}_i} \simeq G_{\mathcal{C}_i} \text{ and } G_{\mathcal{B}_i\cup\mathcal{D}_i} \simeq G_{\mathcal{R}_i\cup\mathcal{C}_i} \text{ for } i \in \{1, 2\}. \tag{4.38}$$

We now show that $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C} = \{-1, 0\}$ and $\mathcal{R}_1 = \mathcal{R}_2 = \mathcal{R} = \{1\}$ satisfy equation 4.38. $G_{\mathcal{R}_i} \simeq G_{\mathcal{B}_i}$ is trivially satisfied as $\mathcal{R}_i$ and $\mathcal{B}_i$ consist of only one particle. Moreover $G_{\mathcal{D}_1}$, $G_{\mathcal{D}_2}$ and $G_{\mathcal{C}}$ are line graphs with two particles while $G_{\mathcal{D}_1\cup\mathcal{B}_1}$, $G_{\mathcal{D}_2\cup\mathcal{B}_2}$ and $G_{\mathcal{R}\cup\mathcal{C}}$ are line graphs with three particles. Any two line graphs with n particles are isomorphic and therefore equation 4.38 is satisfied.

In using our more general formulation we have found a slight improvement in efficiency from the local recursions initially presented for the ring in Section 4.1.2. We had previously needed conditional distributions c_ℓ^t and c_r^t. In our more general formulation, since $\mathcal{C}_i = \mathcal{C}$ and $\mathcal{R}_i = \mathcal{R}$ for $i \in \{1, 2\}$, we need only compute the conditional distribution of $X_{\mathcal{R}}(T)$ given $X_{\mathcal{C}}^T$ since

$$\begin{aligned} c_i^T\left(x_{\mathcal{B}_i}(T), x_{\mathcal{D}_i}^T\right) &= \mathbb{P}\left(X_{\mathcal{B}_i}(T) = x_{\mathcal{B}_i}(T) \;\middle|\; X_{\mathcal{D}_i}^T = x_{\mathcal{D}_i}^T\right) \\ &= \mathbb{P}\left(X_{\mathcal{R}}(T) = x_{\mathcal{B}_i}(T) \;\middle|\; X_{\mathcal{C}}^T = x_{\mathcal{D}_i}^T\right) \end{aligned} \tag{4.39}$$

for $i \in \{1, 2\}$ and so $c_1^T = c_2^T$.

Finally note that (A3) is not satisfied exactly for finite N rings and there exists paths from $\mathcal{B}_1$ to $\mathcal{B}_2$ which do not contain at least two particles in $\mathcal{A}$. One may travel clockwise from $\mathcal{B}_2$ to reach $\mathcal{B}_1$. However, as N grows large the dependence between $\mathcal{B}_1$ and $\mathcal{B}_2$ becomes negligible. Additionally if the ring contains a break then (A3) *is* exactly satisfied. A ring with a break may also be referred to as a line graph or a regular tree with degree 2. In fact, we now show that our local recursions are exact when the interaction network is a regular tree of any finite degree.

4.2.2 d-Regular Trees

A d-regular tree is an acyclic graph where each non-leaf particle has exactly d neighbors. We now express our local recursions on a d-regular tree in terms of our more general formulation and show that they are exact. Without loss of generality we label the root as r with d children $1, 2, ..., d$.

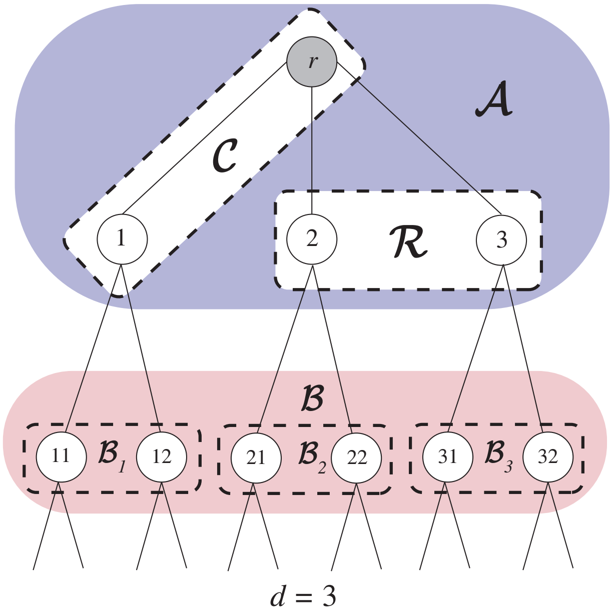

Figure 20: Local Recursions for a d-Regular Tree when $d = 3$

For our local recursions we consider region $\mathcal{A} = \{r, 1, 2, ..., d\}$ with partition $\mathcal{A}_0 = \{r\}$ and $\mathcal{A}_i = \{i\}$ for $i \in \{1, ..., d\}$. We then let $\mathcal{B}_i = \partial i \setminus \{r\}$ for $i \in \{1, ..., d\}$ and $\mathcal{B}_r = \emptyset$. Note that $\mathcal{B}_r, \mathcal{B}_1, ..., \mathcal{B}_d$ forms a disjoint partition of $\partial\mathcal{A} = \mathcal{B}$ and $\partial\mathcal{A}_i \cap \mathcal{B}_j = \emptyset$ for each $i \neq j$ and so (A2) is satisfied. (A3) is also satisfied as each path from $\mathcal{B}_i$ to $\mathcal{B}_j$ must travel through particles $i, r, j \in \mathcal{A}$.

We now claim that $\mathcal{C}_i = \mathcal{C} = \{r, 1\}$ and $\mathcal{R}_i = \mathcal{R} = \{2, ..., d\}$ for $i \in \{1, ..., d\}$ satisfy (A4). Clearly $\mathcal{G}_{B_i} \simeq \mathcal{G}_{\mathcal{R}}$ as $\mathcal{G}_{\mathcal{B}_i}$ and $\mathcal{G}_{\mathcal{R}}$ consist of $d-1$ particles with no interactions. $\mathcal{G}_{\mathcal{D}_i}$ and $\mathcal{G}_{\mathcal{C}}$ are also isomorphic as they are simply line graphs with two particles. Finally we note that both $\mathcal{G}_{\mathcal{B}_i \cup \mathcal{D}_i}$ and $\mathcal{G}_{\mathcal{R} \cup \mathcal{C}}$ are hub and spoke graphs with $d+1$ particles. Therefore they are isomorphic as needed. In a hub and spoke graph the only connections are between a central particle and all other particles, as illustrated by figure 21.

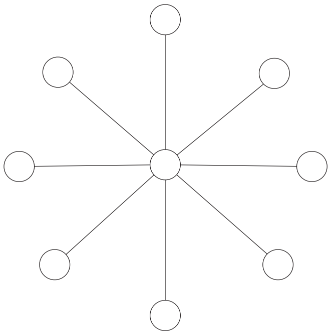

Figure 21: Hub and Spoke Graph with 9 Particles

As a d-regular tree satisfies (A2), (A3) and (A4), our local recursions are exact.

4.2.3 Load Balancing on a Ring

Recall that in the model of load balancing on a ring we observe the interaction network depicted in figure 22. For reference we introduce the model of load balancing in Section 2.2.3.

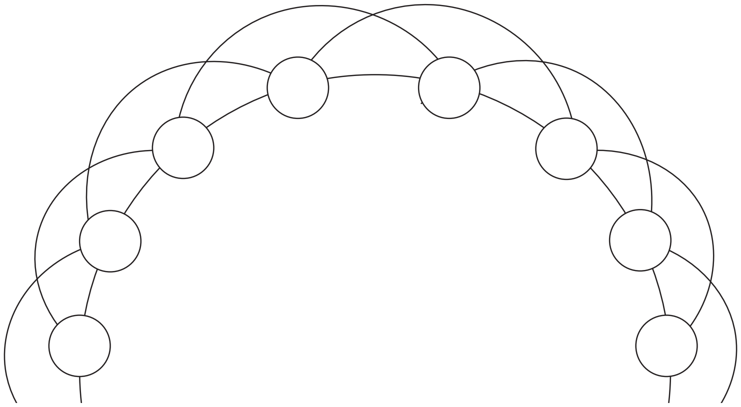

Figure 22: Interaction Network for Load Balancing on a Ring

Some methods we discuss in Section 3 are not applicable for this interaction network, as it contains many short cycles. However, we may still implement our local recursions.

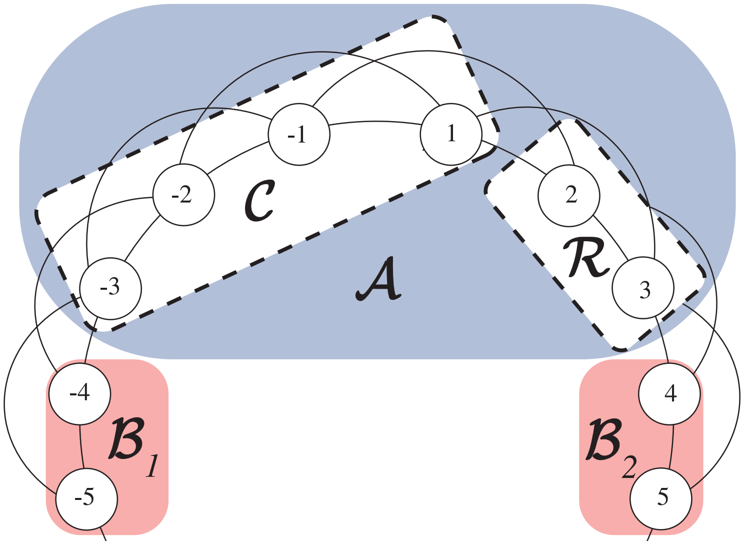

Figure 23: Local Recursions for the Load Balancing Interaction Network

We must now consider a larger local region $\mathcal{A} = \{-3, -2, -1, 1, 2, 3\}$ with partition $\mathcal{A}_1 = \{-3, -2\}$, $\mathcal{A}_2 = \{2, 3\}$, and $\mathcal{A}_3 = \{-1, 1\}$. Both $\mathcal{B}_1$ and $\mathcal{B}_2$ are depicted by figure 23 while $\mathcal{B}_3 = \emptyset$. Once again we find ourselves in the setting where $\mathcal{C}_i = \mathcal{C}$ and $\mathcal{R}_i = \mathcal{R}$ for each i, and so the computation of only one conditional distribution is required for each time.

4.3 The τ-Approximation

Our local recursions, as implemented by algorithm 5, have a computational complexity which is exponential in time T. The exponential complexity arises from the fact that we must consider particle trajectories of length T. For k particles which take values in state space $\mathcal{X}$ there are $|\mathcal{X}|^{kT}$ possible trajectories of length T.

Though this computation is possible for small T, it is infeasible when we wish to understand the long term behavior of a system. As an alternative we now introduce an approximation, which we denote the τ-approximation, that has *linear* computational complexity in T. As we observe in Section 5, the τ-approximation is highly accurate.

In the τ-approximation we consider only the previous τ time steps of the system for some fixed $\tau \in \mathbb{N}$. Where we had previously considered full trajectories

$$X_{\mathcal{A}}^{T} = (X_{\mathcal{A}}(0)x, ..., X_{\mathcal{A}}(T)) \tag{4.40}$$

we now consider τ-trajectories defined as

$$X_{\mathcal{A}}^{\tau,T} = (X_{\mathcal{A}}(T - \tau \vee 0), ..., X_{\mathcal{A}}(T)) \tag{4.41}$$

where $t - \tau \vee 0 = \max(t - \tau, 0)$. Accordingly we aim to compute the distribution $J_{\mathcal{A}}^{\tau,T}$ defined as

$$J_{\mathcal{A}}^{\tau,T}(x_{\mathcal{A}}^{\tau,T}) = \mathbb{P}\left(X_{\mathcal{A}}^{\tau,T} = x_{\mathcal{A}}^{\tau,T}\right). \tag{4.42}$$

The equations for the τ-Approximation are nearly identical to the full local recursions when we replace full trajectories by τ-trajectories. In fact, the τ-Approximation would be exact if the property of conditional independence given the double boundary held for the τ-trajectories, as in equation 4.43.

$$X_{\mathcal{A}}^{\tau,T} \perp\!\!\!\perp X_{\mathcal{V}\setminus(\mathcal{A}\cup\partial^2\mathcal{A})}^{\tau,T} \mid X_{\partial^2\mathcal{A}}^{\tau,T}. \tag{4.43}$$

However, we have seen that in general equation 4.43 is only an approximation when $\tau < T$. Often the full trajectories are required for conditional independence.

In the τ-approximations we replace equations 4.26 and 4.37 of the full local recursions by

$$c_i^{\tau,t}(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^{\tau,t}) = \frac{\sum_{y_A^{\tau,t} \,:\, y_{\mathcal{R}_i}(t)=x_{\mathcal{B}_i}(t),\ y_{\mathcal{C}_i}^{\tau,t}=x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)}{\sum_{y_A^{\tau,t} \,:\, y_{\mathcal{C}_i}^{\tau,t}=x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)} \tag{4.44}$$

and

$$\begin{aligned} J_{\mathcal{A}}^{\tau,t+1}\left(x_{\mathcal{A}}^{\tau,t+1}\right) = \sum_{x_{\mathcal{A}}(t-\tau)} J_{\mathcal{A}}^{\tau,t}\left(x_{\mathcal{A}}^{\tau,t}\right) \prod_{\substack{i\in\{1,\dots,k\} \\ \mathcal{B}_i=\emptyset}} \prod_{v\in\mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \\ * \prod_{\substack{i\in\{1,\dots,k\} \\ \mathcal{B}_i\neq\emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c_i^{\tau,t}\left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^{\tau,t}\right) \prod_{v\in\mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right). \end{aligned} \tag{4.45}$$

In equation 4.45 we introduce the sum $\sum_{x_{\mathcal{A}}(t-\tau)}$ when $\tau \leq t$ as we must consider all possible states of particles in $\mathcal{A}$ at time $t - \tau$. When $\tau > t$ we may ignore the sum. We present an implementation of the τ-approximation below, and Section A.6 of the appendix contains a full derivation.

Algorithm 6 τ-Approximation(T, τ, ν, g)

```
for each $x_{\mathcal{A}}(0)$ do
  $J_{\mathcal{A}}^{\tau,0}\left(x_{\mathcal{A}}(0)\right) = \prod_{v \in \mathcal{A}} \nu(x_v(0))$
end for

for $t = 0, 1, ..., T-1$ do

  {compute each $c_i^{\tau,t}$}
  for $i = 1, 2, ..., k$ do
    if $\mathcal{R}_i = \mathcal{R}_j$ and $\mathcal{C}_i = \mathcal{C}_j$ for some $j < i$ then
      $c_i^{\tau,t} = c_j^{\tau,t}$
    else
      for each $x_{\mathcal{B}_i}(t)$, and $x_{\mathcal{D}_i}^t$ do
        $c_i^{\tau,t}(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^{\tau,t}) = \frac{\sum_{y_A^{\tau,t} \,:\, y_{\mathcal{R}_i}(t)=x_{\mathcal{B}_i}(t),\; y_{\mathcal{C}_i}^{\tau,t}=x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)}{\sum_{y_A^{\tau,t} \,:\, y_{\mathcal{C}_i}^{\tau,t}=x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)}$
      end for
    end if
  end for

  {compute $J_{\mathcal{A}}^{\tau,t+1}$}
  for each $x_{\mathcal{A}}^{\tau,t+1}$ do
    if $\tau \leq t$ then
      $J_{\mathcal{A}}^{\tau,t+1}\left(x_{\mathcal{A}}^{\tau,t+1}\right)$
          $= \sum_{x_{\mathcal{A}}(t-\tau)} J_{\mathcal{A}}^{\tau,t}\left(x_{\mathcal{A}}^{\tau,t}\right) \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i = \emptyset}} \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right)$
          $* \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i \neq \emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c_i^{\tau,t}\left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^{\tau,t}\right) \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right)$
    else
      $J_{\mathcal{A}}^{\tau,t+1}\left(x_{\mathcal{A}}^{\tau,t+1}\right) = J_{\mathcal{A}}^{\tau,t}\left(x_{\mathcal{A}}^{\tau,t}\right) \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i = \emptyset}} \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right)$
          $* \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i \neq \emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c_i^{\tau,t}\left(x_{\mathcal{B}_i}(t), x_{\mathcal{D}_i}^{\tau,t}\right) \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right)$
    end if
  end for
end for
return $J_{\mathcal{A}}^{\tau,T}$
```

4.4 Pair and Triplet-approximations for the Ring

We return to the pair-approximation discussed in Section 3.3.3 with a more analytic perspective. We present some assumptions under which a pair and triplet-approximation are exact for the ring. We also show that algorithm 6 may be used for an implementation of each approximation.

Consider the ring and let $C^{(k)}_{i_1,...,i_k}(t)$ be the proportion of consecutive particles which have states $(i_1, ..., i_k)$. Recall that the Pair-approximation applied by Gast in [6] is

$$C^{(3)}_{i_1,i_2,i_3}(t) = \frac{C^{(2)}_{i_1,i_2}(t) * C^{(2)}_{i_2,i_3}(t)}{C^{(1)}_{i_2}(t)} \tag{4.46}$$

We now consider the hypothetical setting in which the following two conditions hold.

- As the number of particles goes to infinity, $C^{(k)}_{i_1,...,i_k}(t)$ converges to the probability that k consecutive particles will be in states $(i_1, ..., i_k)$ for $k \in \{1, ..., 4\}$.
- $X_u(t)$ is conditionally independent of $X_v(t)$ given $X_{\partial u}(t)$ for $v \notin \partial u$. In other words, the time marginals obey the spatial Markov property.

Under these assumptions the pair-approximation is exact for the ring as

$$\begin{aligned}
&C^{(3)}_{i_1,i_2,i_3}(t) && (4.47)\\
&= \mathbb{P}(X_{u_1}(t) = i_1, X_{u_2}(t) = i_2, X_{u_3}(t) = i_3) && (4.48)\\
&= \mathbb{P}(X_{u_1}(t) = i_1 \mid X_{u_2}(t) = i_2, X_{u_3}(t) = i_3)\\
&\qquad * \mathbb{P}(X_{u_2}(t) = i_2, X_{u_3}(t) = i_3) && (4.49)\\
&= \mathbb{P}(X_{u_1}(t) = i_1 \mid X_{u_2}(t) = i_2)\\
&\qquad * \mathbb{P}(X_{u_2}(t) = i_2, X_{u_3}(t) = i_3) && (4.50)\\
&= \frac{C^{(2)}_{i_1,i_2}(t) * C^{(2)}_{i_2,i_3}(t)}{C^{(1)}_{i_2}(t)}. && (4.51)
\end{aligned}$$

If instead we consider the case where time marginals satisfy the property of conditional independence given the double boundary then a so-called triplet-approximation is exact. We observe that

$$\begin{aligned}
&C^{(4)}_{i_1,i_2,i_3,i_4}(t) && (4.52)\\
&= \mathbb{P}(X_{u_1}(t) = i_1, ..., X_{u_4}(t) = i_4) && (4.53)\\
&= \mathbb{P}(X_{u_1}(t) = i_1 \mid X_{u_2}(t) = i_2, X_{u_3}(t) = i_3, X_{u_4}(t) = i_4)\\
&\qquad * \mathbb{P}(X_{u_2}(t) = i_2, X_{u_3}(t) = i_3, X_{u_4}(t) = i_4) && (4.54)\\
&= \mathbb{P}(X_{u_1}(t) = i_1 \mid X_{u_2}(t) = i_2, X_{u_3}(t) = i_3)\\
&\qquad * \mathbb{P}(X_{u_2}(t) = i_2, X_{u_3}(t) = i_3, X_{u_4}(t) = i_4) && (4.55)\\
&= \frac{C^{(3)}_{i_1,i_2,i_3}(t) * C^{(3)}_{i_2,i_3,i_4}(t)}{C^{(2)}_{i_2,i_3}(t)}. && (4.56)
\end{aligned}$$

The assumption that time marginals satisfy the property of conditional independence given the double boundary corresponds exactly to the τ-approximations we give in Section 4.3 when $\tau = 0$. And so algorithm 6 is an exact implementation of this triplet-approximation with the regions outlined in Section 4.2.1.

We may modify algorithm 6 slightly for an implementation of the pair-approximation on a ring. Consider four consecutive particles $\{-2, -1, 1, 2\}$. We let $\mathcal{A}$ be $\{-1, 1\}$ with $\mathcal{A}_1 = \{-1\}$, $\mathcal{A}_2 = \{1\}$, $\mathcal{B}_1 = \{-2\}$, $\mathcal{B}_2 = \{2\}$, $\mathcal{D}_1 = \{-1\}$ and $\mathcal{D}_2 = \{1\}$. We also let $\mathcal{R}_i = \{-1\}$ and $\mathcal{C}_i = \{1\}$ for $i = 1, 2$.

Finally, we may be able to consider full particle trajectories up to time t and make a more complicated "triplet-approximation". This is a very rough idea which we discuss further in Section A.5 of the appendix.

5 Results

In this section we discuss implementation details and present our results.

5.1 Implementation Details

An efficient implementation of each algorithm discussed in this report may be found in the repository `https://github.com/mwortsma/particle_systems2`. The command line interface and plotting tools are written in Python 2.7 while most of the computation is done by Go (`https://golang.org/`). We chose Go as it is fast and easy to parallelize. Moreover it is easy to compile Go binaries for different systems, enabling us to build locally then run on a computing cluster. For system organization refer to Section A.7 of the appendix.

5.2 Small T Exact Trajectory

We begin by considering the trajectory $X_i^T = (X_i(0), ..., X_i(T))$ of a typical particle. When T is small we may use our local recursions to obtain the the probability of each possible particle trajectory.

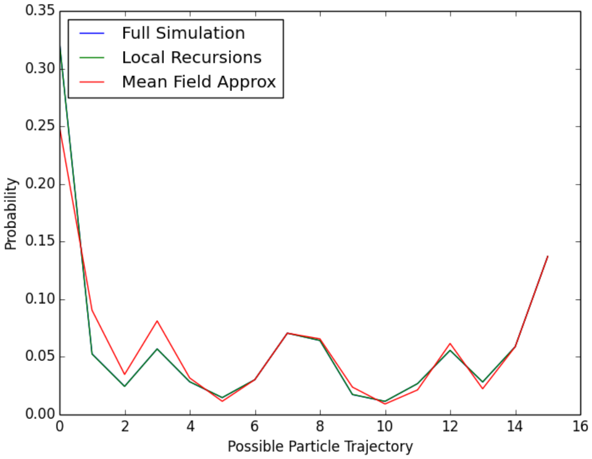

Figure 24: Contact Process on a Ring ($T = 4$, $p = 0.6$, $q = 0.3$)

In figure 25 we examine the contact process on a ring. To estimate the probability of a given trajectory we use $3 * 10^6$ simulations of the full system with $N = 50$ particles. We then predict the probability with the local recursions (algorithm 5) and the mean field approximation (algorithm 3). The local recursions match the full simulation so well that we do not observe any difference from the figures alone. Accordingly we give the relative errors in table 2. We do not expect that they are exactly equivalent as the full simulation is not deterministic. Note that we use the initial conditions $\mathbb{P}(X_i(0) = 0) = 0.6$.

On the y-axis of figure 25 we plot the probability and on the x-axis we have all possible trajectories. Coordinate k on the x-axis corresponds to the trajectory of k converted to binary. For example, coordinate 6 corresponds to trajectory 0110.

Table 1: Relative Error for the Contact Process on a Ring ($T = 4$, $p = 0.6$, $q = 0.3$)

Trajectory	Simulation (Estimated Probability)	Rel. Error (%) (Local Recursions)	Rel. Error (%) (Mean Field)
0 0 0 0	0.322	0.0004	0.2259
0 0 0 1	0.053	0.0031	0.7180
0 0 1 0	0.024	0.0033	0.4226
0 0 1 1	0.057	0.0008	0.4263
0 1 0 0	0.028	0.0036	0.1175
0 1 0 1	0.015	0.0035	0.2168
0 1 1 0	0.030	0.0002	0.0002
0 1 1 1	0.071	0.0007	0.0007
1 0 0 0	0.064	0.0009	0.0226
1 0 0 1	0.017	0.0071	0.3761
1 0 1 0	0.011	0.0057	0.1984
1 0 1 1	0.027	0.0011	0.2039
1 1 0 0	0.056	0.0015	0.1083
1 1 0 1	0.028	0.0044	0.2065
1 1 1 0	0.059	0.0010	0.0010
1 1 1 1	0.137	0.0005	0.0005

We observe in the table above that for certain trajectories the mean field approximation matches our local recursion, even though we are considering the ring. However, we notice that the trajectories which match depend only on the initial conditions. The mean field and local recursions will predict the same initial conditions, as we are using independent initial conditions. For example, consider the trajectory 0110. The first transition $0 \to 1$ depends only on the initial conditions. The next two transitions $1 \to 1$ and $1 \to 0$ occur with probability $1 - q$ and q respectively. For any trajectory which depends on the state of the system at any non-zero time, the mean field approximation will differ from the local recursions.

We now consider the contact process with $T = 5$ on a d-regular tree with $d = 3$. We repeat the experiment with the same system parameters but use initial conditions $\mathbb{P}(X_i(0) = 0) = 0.5$.

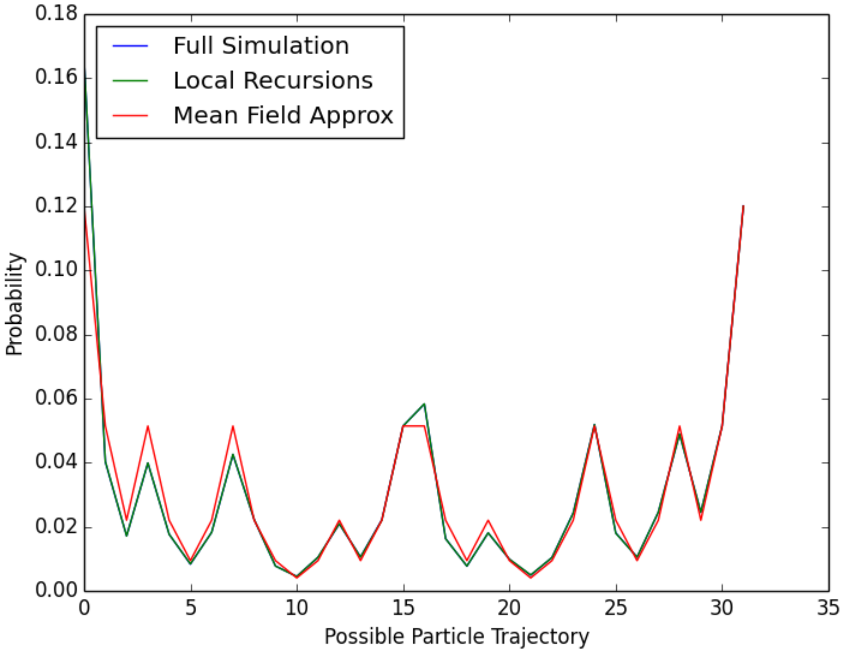

Figure 25: Contact Process on a Tree ($T = 5$, $p = 0.6$, $q = 0.3$)

Notice that the trajectories which depend most on the state of other particles in the system (i.e. 00000, 00001) are the trajectories where the mean field approximation gives the worst prediction.

Table 2: Relative Error for the Contact Process on a Tree ($T = 5$, $p = 0.6$, $q = 0.3$)

Trajectory	Simulation (Estimated Probability)	Rel. Error (%) (Local Recursions)	Rel. Error (%) (Mean Field)
0 0 0 0 0	0.166	0.0002	0.2747
0 0 0 0 1	0.040	0.0030	0.2778
0 0 0 1 0	0.017	0.0012	0.2866
0 0 0 1 1	0.040	0.0010	0.2894
0 0 1 0 0	0.018	0.0011	0.2458
0 0 1 0 1	0.008	0.0093	0.1328
0 0 1 1 0	0.018	0.0013	0.2053
0 0 1 1 1	0.043	0.0021	0.2094
0 1 0 0 0	0.022	0.0015	0.0082
0 1 0 0 1	0.008	0.0040	0.2159
0 1 0 1 0	0.005	0.0037	0.1030
0 1 0 1 1	0.010	0.0022	0.0976
0 1 1 0 0	0.021	0.0030	0.0547
0 1 1 0 1	0.011	0.0048	0.1069
0 1 1 1 0	0.022	0.0057	0.0057
0 1 1 1 1	0.051	0.0006	0.0006
1 0 0 0 0	0.058	0.0011	0.1177
1 0 0 0 1	0.016	0.0031	0.3512
1 0 0 1 0	0.008	0.0041	0.2234
1 0 0 1 1	0.018	0.0001	0.2183
1 0 1 0 0	0.010	0.0029	0.0418
1 0 1 0 1	0.005	0.0127	0.1730
1 0 1 1 0	0.010	0.0023	0.0930
1 0 1 1 1	0.024	0.0042	0.0948
1 1 0 0 0	0.052	0.0004	0.0091
1 1 0 0 1	0.018	0.0050	0.2185
1 1 0 1 0	0.011	0.0012	0.1034
1 1 0 1 1	0.025	0.0011	0.1014
1 1 1 0 0	0.049	0.0005	0.0505
1 1 1 0 1	0.025	0.0014	0.1005
1 1 1 1 0	0.051	0.0001	0.0001
1 1 1 1 1	0.120	0.0008	0.0008

5.3 Approximate Time Marginals

We may use the τ-approximation (algorithm 6) to examine the long term behavior of the system. We focus now on the SIR process and first plot $\mathbb{P}(X_i(t) = \text{Susceptible})$ for $t \in \{0, 1, ..., T\}$.

In the following example, the mean field approximation matches the predicted probability closely for small t. However, as t grows large the mean field approximation predicts that no particle will be left in the susceptible state. In this example we obtain accurate results even when $\tau = 0$. In fact, in example illustrated below, the predicted probability when $\tau = 0$ is indistinguishable from the predicted probability when $\tau = 1$. Though this is not always the case, this result is promising as the $\tau = 0$ approximation is extremely fast.

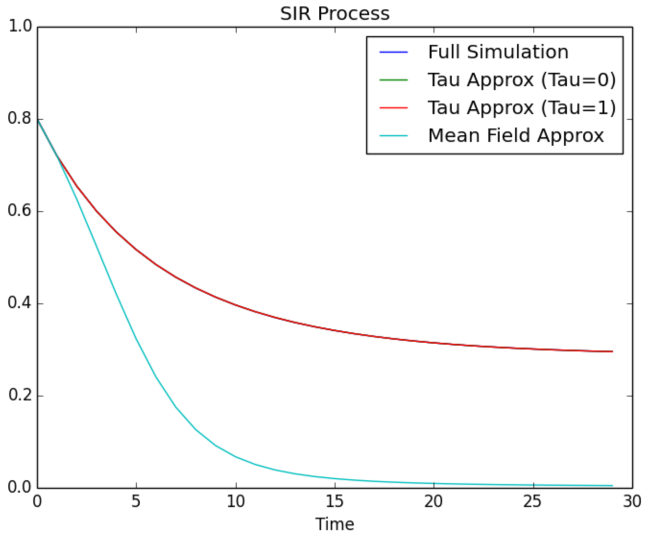

Figure 26: Susceptible Probability for the SIR Process on a Ring ($T = 30$, $p = 0.5$, $q = 0.1$, Initial probability of being susceptible is 0.8)

Table 3: Relative Error for the SIR Process on a Ring ($T = 30$, $p = 0.5$, $q = 0.1$)

Time	Simulation (Estimated Probability of being Susceptible)	Rel. Error (%) (τ-Approx, $\tau = 0$)	Rel. Error (%) (τ-Approx, $\tau = 1$)	Rel. Error (%) (Mean Field)
0	0.80	0.0005	0.0005	0.0005
1	0.72	0.0005	0.0005	0.0005
2	0.65	0.0001	0.0001	0.0429
3	0.60	0.0000	0.0000	0.1271
4	0.55	0.0003	0.0003	0.2434
5	0.52	0.0004	0.0004	0.3751
6	0.48	0.0001	0.0001	0.5040
7	0.46	0.0007	0.0007	0.6171
8	0.43	0.0006	0.0006	0.7085
9	0.41	0.0001	0.0001	0.7783
10	0.40	0.0003	0.0003	0.8297
11	0.38	0.0004	0.0004	0.8674
12	0.37	0.0006	0.0006	0.8948
13	0.36	0.0008	0.0008	0.9150
14	0.35	0.0005	0.0005	0.9300
15	0.34	0.0003	0.0003	0.9414
16	0.33	0.0004	0.0004	0.9501
17	0.33	0.0000	0.0000	0.9568
18	0.32	0.0001	0.0001	0.9621
19	0.32	0.0000	0.0000	0.9663
20	0.31	0.0002	0.0002	0.9698
21	0.31	0.0001	0.0001	0.9725
22	0.31	0.0000	0.0000	0.9748
23	0.31	0.0003	0.0003	0.9767
24	0.30	0.0003	0.0003	0.9783
25	0.30	0.0002	0.0002	0.9796
26	0.30	0.0001	0.0001	0.9808
27	0.30	0.0001	0.0001	0.9818
28	0.30	0.0000	0.0000	0.9826
29	0.30	0.0002	0.0002	0.9833

We believe that the τ-approximations are better when the process converges quickly to an equilibrium. The worst performance we have observed for the τ-approximations is when the process slowly asymptotes. One such example of poor performance from the τ-approximations is illustrated below for the contact process on a ring.

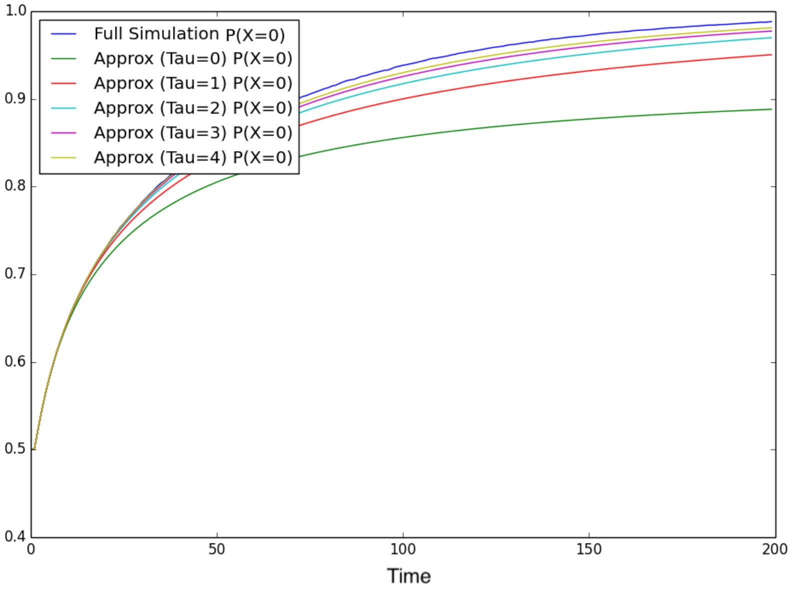

Figure 27: Contact Process on a Ring ($T = 200$, $p = \frac{2}{3}$, $q = \frac{1}{3}$, $\mathbb{P}(X_i(0) = 0) = 0.5$)

From various simulations we make the following conjectures on the behavior of the τ-approximations.

1. $\lim_{\tau\to\infty} J_{\mathcal{A}}^{\tau,T} = J_{\mathcal{A}}^{T}$.

   Moreover if we define $d_0 = J_{\mathcal{A}}^{0,T}$ and $d_i = J_{\mathcal{A}}^{i,T} - J_{\mathcal{A}}^{i-1,T}$ for $i > 0$. Then $\sum_{i=0}^{\infty} d_i = J_{\mathcal{A}}^{T}$.

2. The sequence $\{J_{\mathcal{A}}^{\tau,T}\}_{\tau\in\mathbb{N}}$ is monotone.

3. $\{d_i\}_{i\in\mathbb{N}}$ is a monotone decreasing sequence.

   In fact, it would appear that $\{d_i\}_{i\in\mathbb{N}}$ are decreasing at a rate that is roughly exponential.

5.4 Approximate Equilibrium

We now examine models from statistical physics. We aim to predict the behavior of the system when it has approximately reached equilibrium. For the *parallel Potts* particle system discussed in Section 2.2.5 we expect that the process converges to the Gibbs measure π. When the number of particles is small we may explicitly compute the Gibbs measure, and we may always use MCMC (algorithm 2) to estimate π.

In the following example we consider only 5 particles so that we may explicitly compute π. We let $\beta = 0.5$, $h = J = 1$, and observe that even the $\tau = 0$ approximation of *parallel Potts* converges fairly well to the Gibbs measure. We consider the Potts process where a particle may have state 0,1 or 2. A particle is initialized to state 0,1, or 2 with probability 0.3, 0.3, and 0.4 respectively. From observation we assume that approximated equilibrium is reached at $T = 50$.

We consider the ring and plot the probability of each possible configuration of the local region $\mathcal{A}$ which consists of three consecutive particles at $T = 50$. The MCMC algorithm is slightly more accurate as the τ-approximation implicitly assumes that the ring is infinite or at least large.

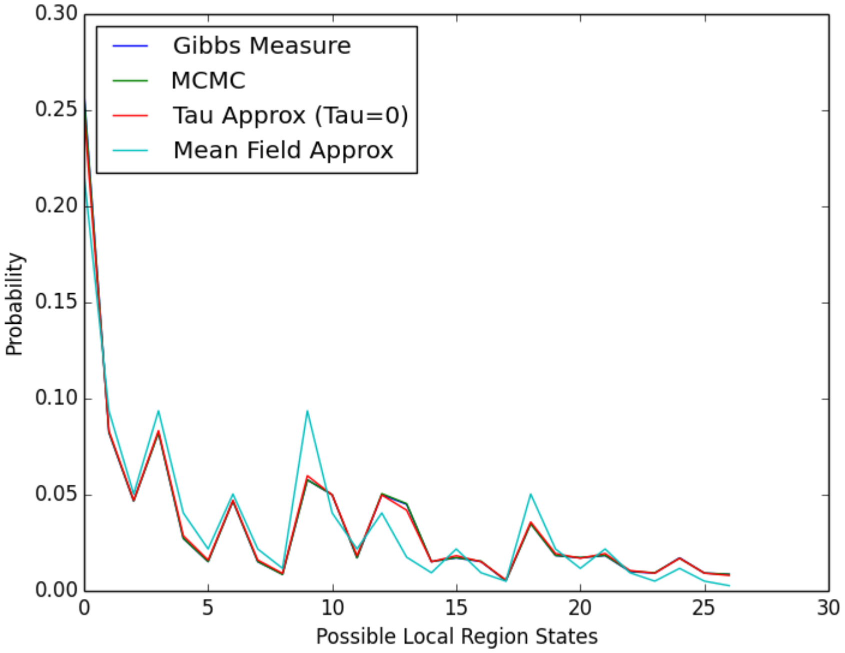

Figure 28: Predicted Approximate Equilibrium for the Potts Process

Table 4: Relative Error for the Potts Process on a Ring

Configuration of Three Consecutive Particles	Probability from Marginalized Gibbs Measure	Rel. Error (%) MCMC (10^6 samples)	Rel. Error (%) (τ-Approx, $\tau = 0$)	Rel. Error (%) (Mean Field)
0 0 0	0.259	0.0059	0.0374	0.1639
0 0 1	0.082	0.0019	0.0112	0.1366
0 0 2	0.047	0.0005	0.0072	0.0776
0 1 0	0.082	0.0022	0.0112	0.1366
0 1 1	0.027	0.0012	0.0508	0.4841
0 1 2	0.015	0.0023	0.0538	0.4313
0 2 0	0.047	0.0014	0.0072	0.0776
0 2 1	0.015	0.0025	0.0538	0.4313
0 2 2	0.009	0.0149	0.0518	0.3540
1 0 0	0.058	0.0009	0.0367	0.6214
1 0 1	0.050	0.0006	0.0023	0.1892
1 0 2	0.017	0.0006	0.0651	0.2675
1 1 0	0.050	0.0113	0.0023	0.1892
1 1 1	0.045	0.0075	0.0676	0.6105
1 1 2	0.015	0.0010	0.0040	0.3807
1 2 0	0.017	0.0060	0.0651	0.2675
1 2 1	0.015	0.0151	0.0040	0.3807
1 2 2	0.005	0.0113	0.0883	0.0341
2 0 0	0.035	0.0049	0.0229	0.4375
2 0 1	0.018	0.0022	0.0557	0.1852
2 0 2	0.017	0.0094	0.0129	0.3185
2 1 0	0.018	0.0040	0.0557	0.1852
2 1 1	0.010	0.0220	0.0527	0.0614
2 1 2	0.009	0.0217	0.0032	0.4509
2 2 0	0.017	0.0121	0.0129	0.3185
2 2 1	0.009	0.0047	0.0032	0.4509
2 2 2	0.009	0.0000	0.0704	0.6850

When n is larger the $\tau = 0$ approximation is even more accurate, as illustrated by the figure below.

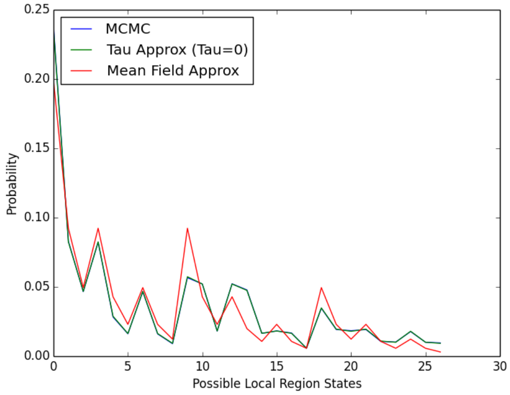

Figure 29: Predicted Approximate Equilibrium for the Potts Process with $n = 20$

5.5 Comparison with the Pair-Approximation

Finally we compare our τ-approximation with the pair-approximation. We first consider the SIR process on a ring with 50 particles. We let $p = 0.9$, $q = 0.1$, and use 10^5 samples of the full simulation to estimate the probability. A particle is initially susceptible with probability 0.8 and infected with probability 0.2. The curves on figure 30 show the predicted probability that a particle is susceptible.

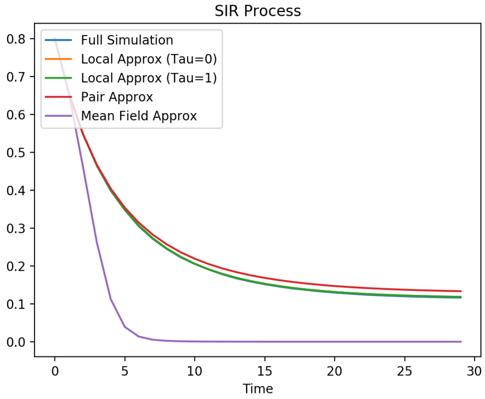

Figure 30: SIR Process Comparison with the Pair-Approximation

We may also simulate the contact process and compare our τ-approximations with the pair-approximation. In figure 31 we use parameters $p = 0.6$, $q = 0.1$ and let a particle be susceptible at time 0 with probability 0.6. $5 * 10^5$ samples of the full simulation are used for an estimate of the probability, and again we show the probability that a particle is susceptible.

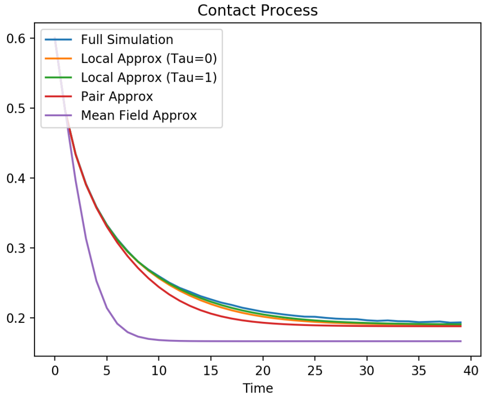

Figure 31: Contact Process Comparison with the Pair-Approximation

As expected our τ-approximation outperforms the pair-approximation. We show the relative errors from figure 29 below.

Table 5: Relative Error for the SIR Process on a Ring ($T = 30$, $p = 0.9$, $q = 0.1$)

Time	Simulation (Estimated Probability of being Susceptible)	Rel. Error (%) (τ-Approx, $\tau = 0$)	Rel. Error (%) (τ-Approx, $\tau = 1$)	Rel. Error (%) (Pair-Approximation)	Rel. Error (%) (Mean Field)
0	0.80	0.0003	0.0003	0.0003	0.0003
1	0.66	0.0021	0.0021	0.0021	0.0021
2	0.55	0.0006	0.0006	0.0006	0.1543
3	0.46	0.0030	0.0030	0.0060	0.4345
4	0.40	0.0055	0.0055	0.0140	0.7183
5	0.35	0.0017	0.0017	0.0177	0.8874
6	0.30	0.0066	0.0066	0.0316	0.9553
7	0.27	0.0062	0.0062	0.0408	0.9804
8	0.24	0.0049	0.0049	0.0495	0.9903
9	0.22	0.0057	0.0057	0.0603	0.9947
10	0.21	0.0037	0.0037	0.0678	0.9969
11	0.19	0.0031	0.0031	0.0762	0.9980
12	0.18	0.0058	0.0058	0.0874	0.9987
13	0.17	0.0089	0.0089	0.0983	0.9990
14	0.16	0.0080	0.0080	0.1040	0.9993
15	0.15	0.0082	0.0082	0.1101	0.9995
16	0.15	0.0087	0.0087	0.1157	0.9996
17	0.14	0.0110	0.0110	0.1226	0.9997
18	0.14	0.0091	0.0091	0.1240	0.9997
19	0.13	0.0121	0.0121	0.1303	0.9998
20	0.13	0.0131	0.0131	0.1339	0.9998
21	0.13	0.0137	0.0137	0.1366	0.9998
22	0.12	0.0160	0.0160	0.1407	0.9998
23	0.12	0.0168	0.0168	0.1427	0.9999
24	0.12	0.0170	0.0170	0.1439	0.9999
25	0.12	0.0173	0.0173	0.1449	0.9999
26	0.12	0.0182	0.0182	0.1465	0.9999
27	0.12	0.0181	0.0181	0.1467	0.9999
28	0.12	0.0182	0.0182	0.1470	0.9999
29	0.12	0.0170	0.0170	0.1457	0.9999

6 Conclusion and Future Work

In this report we have presented local recursions which allow for the exact computation of marginal distributions for interacting particle systems on sparse graphs. We have demonstrated the effectiveness of our recursions when T is small. Moreover, we have motivated a fast approximation which may be used to understand the long term behavior and approximate equilibrium.

Additionally, we have discussed a series of important examples of interacting particle systems and surveyed previous work which aims to understand the dynamics of a typical particle. With novel theory introduced by [11] we have been able to examine existing theory with a new perspective.

This work is only the beginning. There is vast opportunity for the application of [11] to understand systems where analytic solutions were previously thought to be out of reach. We have not ventured far beyond discrete time in this report, and we ask if there is an equivalent formulation of our local recursions for continuous time. Moreover, we wonder if the local recursions may be made to be more efficient. Perhaps in certain cases there is some τ where the τ-approximation is exact. We already observe that in certain settings we achieve accurate results even for $\tau = 0$. As discussed in Section 4.4, if the approximation when $\tau = 0$ is exact then we may use a triplet-approximation to write down a series of exact differential equations for the ring.

Finally, we believe that there is a myriad of applications for our local recursions and conditional independence given the double boundary that we would like to explore. For instance, we are interested in what results we could achieve if we were to model a pixel in a video as a particle trajectory. Geman and Geman invent Gibbs sampling for the bayesian restoration of images in [7]. We wonder if we could condition on the double boundary instead and consider the bayesian restoration of videos. In computer vision there has also been frequent use of belief propagation algorithms for static images [5]. Perhaps we may use our discussions in Section 3.2 to employ similar techniques for videos.

Interacting particle systems will continue to be an interesting area of study for many years to come and we look forward to future work.

A Appendix

A.1 Proof of Lemma 2.1.

Since A, B and S are disjoint sets and every path from A to B contains at least one node in S it follows that $\partial A \cap B = \emptyset$. We then have

$$\begin{aligned}
&\mathbb{P}(X_A = x_A | X_S = x_S) && \text{(A.1)}\\
&= \sum_{x_B} \mathbb{P}(X_A = x_A | X_S = x_S, X_B = x_B)\mathbb{P}(X_B = x_B | X_S = x_S) && \text{(A.2)}\\
&= \sum_{x_B} \mathbb{P}(X_A = x_A | X_{\mathcal{V}\setminus A} = x_{\mathcal{V}\setminus A})\mathbb{P}(X_B = x_B | X_S = x_S) \qquad \text{as } B \cup S = \mathcal{V} \setminus A && \text{(A.3)}\\
&= \sum_{x_B} \mathbb{P}(X_A = x_A | X_{\partial A} = x_{\partial A})\mathbb{P}(X_B = x_B | X_S = x_S) && \text{(A.4)}\\
&= \mathbb{P}(X_A = x_A | X_{\partial A} = x_{\partial A}) \sum_{x_B} \mathbb{P}(X_B = x_B | X_S = x_S) \qquad \text{as } \partial A \cap B = \emptyset && \text{(A.5)}\\
&= \mathbb{P}(X_A = x_A | X_{\partial A} = x_{\partial A}) && \text{(A.6)}\\
&= \mathbb{P}(X_A = x_A | X_{\mathcal{V}\setminus A} = x_{\mathcal{V}\setminus A}) && \text{(A.7)}\\
&= \mathbb{P}(X_A = x_A | X_S = x_S, X_B = x_B) \qquad \text{as } B \cup S = \mathcal{V} \setminus A && \text{(A.8)}
\end{aligned}$$

□

A.2 Proof of Remark 2.1

We claim that

$$X_i(t) = f(X_i(t-1), X_{\partial i}(t-1), \xi_i(t)) \tag{A.9}$$

and

$$\mathbb{P}(X_i(t) = v \mid X_i(t-1) = u) = g(v, u, X_{\partial i}(t-1)) \tag{A.10}$$

are equivalent formulations. We will first show that A.9 implies A.10 and then that A.10 implies A.9.

First consider the case where we have some function f which satisfies equation A.9. Notice that for a given u, v

$$\mathbb{P}(X_i(t) = v \mid X_i(t-1) = u) \tag{A.11}$$

$$= \mathbb{P}(f(u, X_{\partial i}(t-1), \xi_i(t)) = v) \tag{A.12}$$
$$= \mathbb{P}(f(u, x_{\partial i}, \xi_i(t)) = v | X_{\partial i}(t-1) = x_{\partial i})\mathbb{P}(X_{\partial i}(t-1) = x_{\partial i}) \tag{A.13}$$

and there exists some $A \in \mathbb{R}$ such that A.13 becomes

$$\mathbb{P}(\xi_i(t) \in A | X_{\partial i}(t-1) = x_{\partial i})\mathbb{P}(X_{\partial i}(t-1) = x_{\partial i}) \tag{A.14}$$
$$= \mathbb{P}(\xi_i(t) \in A)\mathbb{P}(X_{\partial i}(t-1) = x_{\partial i}) \qquad \text{by independence.} \tag{A.15}$$

So we may define $g(v, u, x_{\partial i}) = \mathbb{P}(\xi_i(t) \in A)\mathbb{P}(X_{\partial i}(t-1) = x_{\partial i})$ and g will satisfy A.10.

Now we consider some g which satisfies A.10 and show that we may construct a function f which satisfies A.9.

Let $\xi_i(t)$ be uniform on the interval [0,1]. Now fix $u, x_{\partial i}$ and consider the k possible values for $X_i(t)$ denoted as $\{v_1, ..., v_k\}$. Divide the interval [0,1] into k disjoint intervals $A_1, ..., A_k$ where A_i has length $g(v_i, u, x_{\partial i})$. Then define

$$f(u, x_{\partial i}, \xi_i(t)) = \sum_{i=1}^{k} \delta_{A_i}(\xi_i(t))v_i \tag{A.16}$$

where $\delta_{A_i}(\xi_i(t))$ is 1 if $\xi_i(t) \in A_i$ and 0 otherwise.

A.3 Proof of Lemma 2.2

Consider four adjacent particles i, j, k, l on a line graph with as depicted by figure 32.

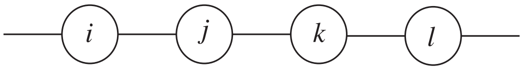

Figure 32: Four Particles on a Line Graph

We consider the following process which we call the *modified exclusion process*, defined by equations A.17, A.18 and A.19.

$$\{X_v(0)\}_{v \in \mathcal{V}} \sim \text{i.i.d. Bernoulli}(\delta) \tag{A.17}$$

$$\{\xi_v(t)\}_{v \in \mathcal{V}, t \in \mathbb{T}} \sim \text{i.i.d. Bernoulli}(\epsilon) \tag{A.18}$$

For any consecutive particles a, b and c,

$$X_b(t) = \max \begin{cases} X_b(t-1) \\ X_a(t-1) \oplus X_c(t-1) \\ \xi_b(t) \end{cases} \tag{A.19}$$

for $t > 0$ where $\oplus$ is the XOR operator.

Note that if $X_v(t) = 1$ then $X_b(s) = 1$ for any $s > t$. We are introducing noise into the system so that we are are not considering configurations with 0 probability.

It is possible to see that the trajectories do not obey the spatial Markov property by examining the times $\mathbb{T} = \{0, 1\}$. If the trajectories did obey the spatial Markov property then we would have

$$\begin{aligned} &\mathbb{P}(X_j^1 = x_j^1 | X_i^1 = x_i^1, X_k^1 = x_k^1, X_l^1 = x_l^1) \\ &= \mathbb{P}(X_j^1 = x_j^1 | X_i^1 = x_i^1, X_k^1 = x_k^1) \end{aligned} \tag{A.20}$$

as $l \cap \partial j = \emptyset$.

If $X_k^1 = (0, 1)$ and both $X_j(0)$ and $X_l(0)$ are 0 it follows immediately that $\xi_k(1) = 1$. Consequently we have

$$\mathbb{P}(X_j^1 = (0,0) | X_i^1 = (0,0), X_k^1 = (0,1), X_l^1 = (0,0)) \tag{A.21}$$
$$= \mathbb{P}(X_j(0) = 0 \cap \xi_j(1) = 0 \cap \xi_k(1) = 1) \tag{A.22}$$
$$= \mathbb{P}(X_j(0) = 0)\mathbb{P}(\xi_j(1) = 0)\mathbb{P}(\xi_k(1) = 1) \tag{A.23}$$
$$= (1 - \delta)(1 - \epsilon)\epsilon \tag{A.24}$$

However,

$$\mathbb{P}(X_j^1 = (0,0) | X_i^1 = (0,0), X_k^1 = (0,1)) \tag{A.25}$$
$$= \mathbb{P}\left(X_j(0) = 0 \cap \xi_j(1) = 0 \cap (\{\xi_k(1) = 1\} \cup \{X_l(0) = 1\})\right) \tag{A.26}$$
$$= \mathbb{P}(X_j(0) = 0)\mathbb{P}(\xi_j(1) = 0)\mathbb{P}\left(\{\xi_k(1) = 1\} \cup \{X_l(0) = 1\}\right) \tag{A.27}$$
$$= (1 - \delta)(1 - \epsilon)(\epsilon + \delta - \epsilon\delta) \tag{A.28}$$

and so the trajectories do not obey the spatial Markov property.

We now give a similar counterexample for the time marginals $X_u(t)$.

If the time marginals $X_u(t)$ did obey the spatial Markov property then we would have that

$$\begin{aligned} &\mathbb{P}(X_j(1) = x_j | X_i(1) = x_i, X_k(1) = x_k, X_l(1) = x_l) \\ &= \mathbb{P}(X_j(1) = x_j | X_i(1) = x_i, X_k(1) = x_k). \end{aligned} \tag{A.29}$$

However we notice

$$\mathbb{P}(X_j(1) = 0 | X_i(1) = 0, X_k(1) = 1, X_l(1) = 0) \tag{A.30}$$
$$= \mathbb{P}\left(X_j(0) = 0 \cap \xi_j(1) = 0 \cap (\{\xi_k(1) = 1\} \cup \{X_k(0) = 1\})\right) \tag{A.31}$$
$$= \mathbb{P}(X_j(0) = 0)\mathbb{P}(\xi_j(1) = 0)\mathbb{P}\left(\{\xi_k(1) = 1\} \cup \{X_k(0) = 1\}\right) \tag{A.32}$$
$$= (1 - \delta)(1 - \epsilon)(\epsilon + \delta - \epsilon\delta) \tag{A.33}$$

and

$$\mathbb{P}(X_j(1) = 0 | X_i(1) = 0, X_k(1) = 1) \tag{A.34}$$
$$= \mathbb{P}\left(X_j(0) = 0 \cap \xi_j(1) = 0 \cap (\{\xi_k(1) = 1\} \cup \{X_k(0) = 1\} \cup \{X_l(0) = 1\})\right) \tag{A.35}$$
$$= \mathbb{P}(X_j(0) = 0)\mathbb{P}(\xi_j(1) = 0)\mathbb{P}\left(\{\xi_k(1) = 1\} \cup \{X_k(0) = 1\} \cup \{X_l(0) = 1\}\right) \tag{A.36}$$
$$= (1-\delta)(1-\epsilon)(\epsilon + 2\delta - \epsilon\delta - \delta^2 + \epsilon\delta^2) \tag{A.37}$$

and so both the time marginals $X_i(t)$ and the complete trajectories X_i^T need not obey the spatial Markov property. □

A.4 Proof of Lemma 2.5.

Since A, B and S are disjoint sets and every path from A to B contains at least two particles in S it follows that $\partial^2 A \cap B = \emptyset$. We simplify the notation so that X_H is taken to mean X_H^T for any H. We then have

$$\mathbb{P}(X_A = x_A | X_S = x_S) \tag{A.38}$$
$$= \sum_{x_B} \mathbb{P}(X_A = x_A | X_S = x_S, X_B = x_B)\mathbb{P}(X_B = x_B | X_S = x_S) \tag{A.39}$$
$$= \sum_{x_B} \mathbb{P}(X_A = x_A | X_{\mathcal{V}\setminus A} = x_{\mathcal{V}\setminus A})\mathbb{P}(X_B = x_B | X_S = x_S) \qquad \text{as } B \cup S = \mathcal{V} \setminus A \tag{A.40}$$
$$= \sum_{x_B} \mathbb{P}(X_A = x_A | X_{\partial^2 A} = x_{\partial^2 A})\mathbb{P}(X_B = x_B | X_S = x_S) \tag{A.41}$$
$$= \mathbb{P}(X_A = x_A | X_{\partial^2 A} = x_{\partial^2 A}) \sum_{x_B} \mathbb{P}(X_B = x_B | X_S = x_S) \qquad \text{as } \partial^2 A \cap B = \emptyset \tag{A.42}$$
$$= \mathbb{P}(X_A = x_A | X_{\partial^2 A} = x_{\partial A}) \tag{A.43}$$
$$= \mathbb{P}(X_A = x_A | X_{\mathcal{V}\setminus A} = x_{\mathcal{V}\setminus A}) \tag{A.44}$$
$$= \mathbb{P}(X_A = x_A | X_S = x_S, X_B = x_B) \qquad \text{as } B \cup S = \mathcal{V} \setminus A \tag{A.45}$$

□

A.5 The "Triplet" Approximation for the SIR Process on a Ring (Full Particle Trajectories)

Consider the SIR process on a ring. We say that particle j has trajectory $u_{i,r}$ if particle j is susceptible for $t \in [0, i)$, infected for $t \in [i, r)$, and recovered for $t \in$

$[r, \infty)$. Let $C^{(k)}_{(i_1,r_1),...,(i_k,r_k)}$ be the proportion of consecutive particles $a_1, ..., a_k$ such that particle a_j has trajectory u_{i_j,r_j}.

We use conditional independence given the double boundary to show that a variation of the triplet-approximation is exact as the number of particles tends to infinity and the following assumption holds: For $k \in \{1, ..., 4\}$ $C^{(k)}_{(i_1,r_1),...,(i_k,r_k)}$ converges to $\mathbb{P}\left(X^T_{a_1} = u_{i_1,j_1}, ..., X^T_{a_k} = u_{i_k,j_k}\right)$ for consecutive particles $a_1, ..., a_k$. Now observe that

$$C^{(4)}_{(i_1,r_1),...,(i_4,r_4)} = \frac{C^{(3)}_{(i_1,r_1),...,(i_3,r_3)} * C^{(3)}_{(i_2,r_2),...,(i_4,r_4)}}{C^{(2)}_{(i_2,r_2),(i_3,r_3)}} \tag{A.46}$$

exactly as

$$\begin{aligned}
& C^{(4)}_{(i_1,r_1),...,(i_4,r_4)} && \text{(A.47)}\\
&= \mathbb{P}\left(X^T_{a_1} = u_{i_1,j_1}, ..., X^T_{a_4} = u_{i_4,j_4}\right) && \text{(A.48)}\\
&= \mathbb{P}\left(X^T_{a_1} = u_{i_1,j_1} \mid X^T_{a_2} = u_{i_2,j_2},\ X^T_{a_3} = u_{i_3,j_3},\ X^T_{a_4} = u_{i_4,j_4}\right) \\
&\qquad * \mathbb{P}\left(X^T_{a_2} = u_{i_2,j_2},\ X^T_{a_3} = u_{i_3,j_3},\ X^T_{a_4} = u_{i_4,j_4}\right) && \text{(A.49)}\\
&= \mathbb{P}\left(X^T_{a_1} = u_{i_1,j_1} \mid X^T_{a_2} = u_{i_2,j_2},\ X^T_{a_3} = u_{i_3,j_3}\right) \\
&\qquad * \mathbb{P}\left(X^T_{a_2} = u_{i_2,j_2},\ X^T_{a_3} = u_{i_3,j_3},\ X^T_{a_4} = u_{i_4,j_4}\right) && \text{(A.50)}\\
&= \frac{C^{(3)}_{(i_1,r_1),...,(i_3,r_3)} * C^{(3)}_{(i_2,r_2),...,(i_4,r_4)}}{C^{(2)}_{(i_2,r_2),(i_3,r_3)}} && \text{(A.51)}
\end{aligned}$$

where A.50 follows from the property of conditional dependence given the double boundary. And so, theoretically, we may write down a series of exact differential equations for the SIR Process on a ring. This is only a very rough idea which we believe may warrant future work.

A.6 Full Derivation of the τ-Approximation

In deriving the τ-approximation we assume the property of conditional independence given the double boundary of the τ-trajectories alone.

$$X^{\tau,T}_{\mathcal{A}} \perp\!\!\!\perp X^{\tau,T}_{\mathcal{V}\setminus(\mathcal{A}\cup\partial^2 A)} \mid X^{\tau,T}_{\partial^2\mathcal{A}}. \tag{A.52}$$

We now show that under the assumptions (A1) through (A4) and equation A.52 we may compute $J^{\tau,t+1}_{\mathcal{A}}$ given $J^{\tau,t}_{\mathcal{A}}$ and $\{c^{\tau,t}_i\}_{i=1,..,k}$, where $c^{\tau,t}_i$ satisfies

$$\begin{aligned}
& c^{\tau,t}_i\left(x_{\mathcal{B}_i}(t), x^t_{\mathcal{D}_i}\right) && \text{(A.53)}\\
&= \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X^{\tau,t}_{\mathcal{D}_i} = x^{\tau,t}_{\mathcal{D}_i}\right) && \text{(A.54)}
\end{aligned}$$

$$= \mathbb{P}\left(X_{\mathcal{R}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X_{\mathcal{C}_i}^{\tau,t} = x_{\mathcal{D}_i}^{\tau,t}\right) \tag{A.55}$$

$$= \frac{\sum_{y_{\mathcal{A}}^{\tau,t} \,:\, y_{\mathcal{R}_i}(t) = x_{\mathcal{B}_i}(t),\; y_{\mathcal{C}_i}^{\tau,t} = x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)}{\sum_{y_{\mathcal{A}}^{\tau,t} \,:\, y_{\mathcal{C}_i}^{\tau,t} = x_{\mathcal{D}_i}^{\tau,t}} J_{\mathcal{A}}^{\tau,t}\left(y_{\mathcal{A}}^{\tau,t}\right)} \tag{A.56}$$

for $\mathbb{P}(X_{\mathcal{C}_i}^{\tau,t} = x_{\mathcal{D}_i}^{\tau,t}) > 0$.

We assume now that $\tau \leq t$ as the case where $\tau > t$ is handled Lemma 4.3. We observe that for any $x_{\mathcal{A}}^{\tau,t+1}$ we have

$$J_{\mathcal{A}}^{\tau,t+1}\left(x_{\mathcal{A}}^{\tau,t+1}\right) = \mathbb{P}\left(X_{\mathcal{A}}^{\tau,t+1} = x_{\mathcal{A}}^{\tau,t+1}\right) \tag{A.57}$$

$$= \sum_{x_{\mathcal{A}}(t-\tau)} \mathbb{P}\left(X_{\mathcal{A}}^{\tau,t+1} = x_{\mathcal{A}}^{\tau,t+1},\; x_{\mathcal{A}}(t-\tau)\right) \tag{A.58}$$

$$= \sum_{x_{\mathcal{A}}(t-\tau)} \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right) \mathbb{P}\left(X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right). \tag{A.59}$$

As $\mathbb{P}\left(X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right)$ is simply $J_{\mathcal{A}}^{\tau,t}\left(x_{\mathcal{A}}^{\tau,t}\right)$ which we have by assumption, we are now left to compute $\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \middle| X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right)$. We notice that

$$\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \middle| X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right) \tag{A.60}$$

$$\begin{aligned} &= \sum_{x_{\mathcal{B}}(t)} \mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t},\; X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t)\right) \\ &\qquad * \mathbb{P}\left(X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right) \end{aligned} \tag{A.61}$$

and now examine the two terms in the summation above separately. We may simplify the first term as

$$\begin{aligned} &\mathbb{P}\left(X_{\mathcal{A}}(t+1) = x_{\mathcal{A}}(t+1) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t},\; X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t)\right) \\ &= \prod_{v \in \mathcal{A}} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \end{aligned} \tag{A.62}$$

Finally we use conditional independence of the τ-trajectories given the double boundary (equation A.52) to simply the latter term.

$$\mathbb{P}\left(X_{\mathcal{B}}(t) = x_{\mathcal{B}}(t) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right) \tag{A.63}$$

$$= \prod_{i=1}^{k} \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X_{\mathcal{A}}^{\tau,t} = x_{\mathcal{A}}^{\tau,t}\right) \tag{A.64}$$

$$= \prod_{i=1}^{k} \mathbb{P}\left(X_{\mathcal{B}_i}(t) = x_{\mathcal{B}_i}(t) \;\middle|\; X^{\tau,t}_{\mathcal{D}_i} = x^{\tau,t}_{\mathcal{D}_i}\right) \tag{A.65}$$

$$= \prod_{i=1}^{k} c_i^{\tau,t}\left(x_{\mathcal{B}_i}(t), x^{\tau,t}_{\mathcal{D}_i}\right) \tag{A.66}$$

In summary we have

$$\begin{aligned} &J^{\tau,t+1}_{\mathcal{A}}\left(x^{\tau,t+1}_{\mathcal{A}}\right) \\ &= \sum_{x_{\mathcal{A}}(t-\tau)} J^{t}_{\mathcal{A}}\left(x^{\tau,t}_{\mathcal{A}}\right) \sum_{x_{\mathcal{B}}(t)} \prod_{v \in \mathcal{A}} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \prod_{i=1}^{k} c_i^{t}\left(x_{\mathcal{B}_i}(t), x^{\tau,t}_{\mathcal{D}_i}\right) \end{aligned} \tag{A.67}$$

which we may factor to obtain

$$\begin{aligned} &J^{\tau,t+1}_{\mathcal{A}}\left(x^{\tau,t+1}_{\mathcal{A}}\right) \\ &= \sum_{x_{\mathcal{A}}(t-\tau)} J^{\tau,t}_{\mathcal{A}}\left(x^{\tau,t}_{\mathcal{A}}\right) \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i = \emptyset}} \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right) \\ &\qquad * \prod_{\substack{i \in \{1,\ldots,k\} \\ \mathcal{B}_i \neq \emptyset}} \sum_{x_{\mathcal{B}_i}(t)} c_i^{\tau,t}\left(x_{\mathcal{B}_i}(t), x^{\tau,t}_{\mathcal{D}_i}\right) \prod_{v \in \mathcal{A}_i} g\left(x_v(t+1), x_v(t), x_{\partial v}(t)\right). \end{aligned} \tag{A.68}$$

A.7 System Organization

Our system is organized as follows. The `cmd2` Go binary may be run to obtain the results of a single algorithm. For example, the command

```
cmd2 -contact_graph_time -graph=ring -T=30 -p=0.6 -q=0.1 \
 -n=50 -steps=10000 -nu=[0.6,0.4]
```

will use 10000 samples of the full simulation of the contact process with 50 particles to estimate $\mathbb{P}(X_i(t) = k)$ at each time $t \in [0, 30]$ with the parameters $p = 0.6$, $q = 0.1$, and the initial conditions $\mathbb{P}(X_i(0) = 0) = 0.6$, $\mathbb{P}(X_i(0) = 1) = 0.4$. We may change the graph type to `-graph=complete` for the complete graph or `-graph=er` for an Erdős-Rényi graph.

If we wish to compare different algorithms we use the python command line interface given by `main.py`.

```
python main.py \
-commands="<command 1> | <command 2> | ... | <command n>" \
-shared="<arguments shared between each command> \
-show_plot -type=<plot type>
```

To compare the full simulation above to the τ-approximation ($\tau = 1$) and the mean field approximation we run

```
python main.py \
 -commands="cmd2 -contact_graph_time -graph=ring -n=50 -steps=20000 |\
            cmd2 -contact_local_time -tau=1 -d=2 |\
            cmd2 -contact_meanfield_time"\
 -shared="-T=30 -p=0.6 -q=0.1 -nu=[0.6,0.4]"\
 -show_plot -type=time\
 -labels="Full Simulation, Local Approx (Tau=1), Mean Field Approx"\
 -title="Contact Process"
```

which generates the following figure.

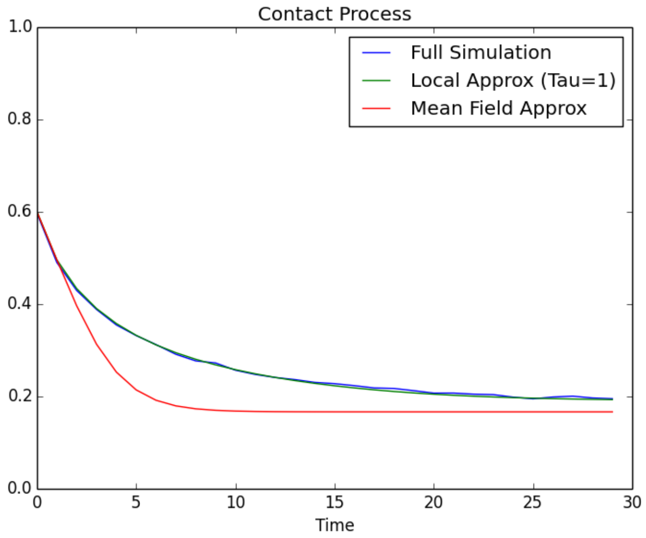

Figure 33: Example Generated by the Above Command

If we are instead interested in the path of a typical particle, we may enumerate each possible particle path on the x-axis. Note that we may only do this for small values of T. For small values of T we may also omit the τ argument as τ will default to ∞ (and so we will recover the full local recursions).

```
python main.py \
 -commands="cmd2 -contact_graph_path -graph=ring -n=50 -steps=20000 |\
            cmd2 -contact_local_path  -d=2 |\
            cmd2 -contact_meanfield_path"\
 -shared="-T=4 -p=0.6 -q=0.1 -nu=[0.6,0.4]"\
 -show_plot -type=path\
 -labels="Full Simulation, Local Recursions, Mean Field Approx"\
 -title="Contact Process"
```

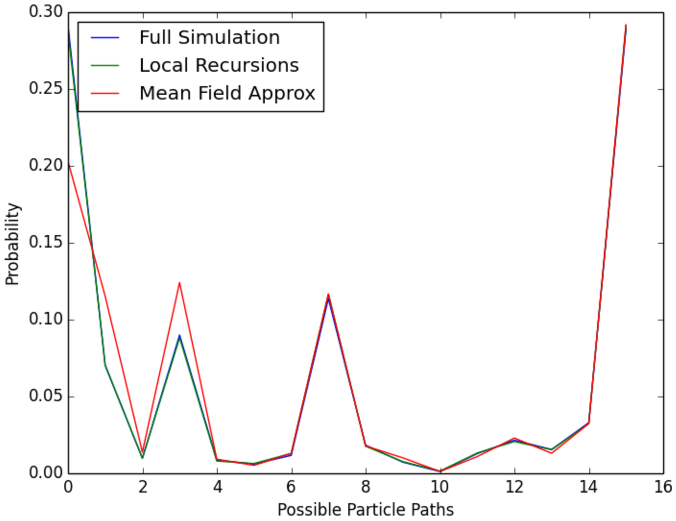

Figure 34: Example Generated by the Above Command

References

[1] Erik Aurell and Hamed Mahmoudi. A message-passing scheme for non-equilibrium stationary states. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(04):P04014, 2011.

[2] Erik Aurell and Hamed Mahmoudi. Three lemmas on dynamic cavity method. *Communications in Theoretical Physics*, 56(1):157, 2011.

[3] Hans Bethe and Henry Herbert Wills. Statistical theory of superlattices. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 150(871):552–575, 1935.

[4] Shankar Bhamidi, Amarjit Budhiraja, and Ruoyu Wu. Weakly interacting particle systems on inhomogeneous random graphs, 2016.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, Oct 2006.

[6] Nicolas Gast. The Power of Two Choices on Graphs: the Pair-Approximation is Accurate. In *Workshop on MAthematical performance Modeling and Analysis*, Portland, United States, June 2015.

[7] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, Nov 1984.

[8] J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.

[9] Yashodhan Kanoria and Andrea Montanari. Majority dynamics on trees and the dynamic cavity method. *Ann. Appl. Probab.*, 21(5):1694–1748, 10 2011.

[10] Daphne Koller, Nir Friedman, Lise Getoor, and Benjamin Taskar. Graphical models in a nutshell. In L. Getoor and B. Taskar, editors, *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[11] Daniel Lacker, Kavita Ramanan, and Ruoyu Wu. Preprint, 2018.

[12] Thomas M. Liggett. Stochastic models of interacting systems. *Ann. Probab.*, 25(1):1–29, 01 1997.

[13] Andrey Y. Lokhov. *Dynamic cavity method and problems on graphs*. Theses, Université Paris Sud - Paris XI, November 2014.

[14] Andrey Y. Lokhov, Marc Mzard, and Lenka Zdeborov. Dynamic message-passing equations for models with unidirectional dynamics. 2014.

[15] Maia Martcheva. *Introduction to Epidemic Modeling*, pages 9–31. Springer US, Boston, MA, 2015.

[16] Michael Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, October 2001.

[17] Karl Oelschlager. A martingale approach to the law of large numbers for weakly interacting stochastic processes. *Ann. Probab.*, 12(2):458–479, 05 1984.

[18] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[19] Robin Pemantle. The contact process on trees. *Ann. Probab.*, 20(4):2089–2116, 10 1992.

[20] Kavita Ramanan. Graphical models and their applications. Course Notes, Personal Communication, 2012.

[21] Nikkala A. Thomson and Stephen P. Ellner. Pair-edge approximation for heterogeneous lattice population models. *Theoretical Population Biology*, 64(3):271 – 280, 2003. Understanding the role of environmental vatiation in population and community dynamics.