Controlling Heterogeneous Stochastic Growth Processes on Lattices with Limited Resources

Ravi N. Haksar[1], Friedrich Solowjow[2], Sebastian Trimpe[2], and Mac Schwager[3]

***Abstract*— We consider controlling a heterogeneous stochastic growth process defined on a lattice with a control resource constraint. We address heterogeneous effects in three respects: (i) the process grows at different rates for different directions on the lattice, (ii) the nodes of the lattice may have different dynamics, and (iii) nodes may have different priorities for control. We use a forest wildfire driven by a west-to-east wind near an urban region to illustrate our approach, where preserving the urban region is prioritized over the forest. We leverage the Galton-Watson branching process as an approximation to predict the process growth rate and stopping time and to construct effective control policies. Our approach is also applicable to processes with an underlying graph structure, such as robot swarms, disease epidemics, computer viruses, and social networks. In contrast to prior work, we directly address heterogeneous models and our framework allows for a broader class of control policy descriptions. Lastly, we characterize the conditions under which a control policy will stabilize a supercritical heterogeneous growth process.**

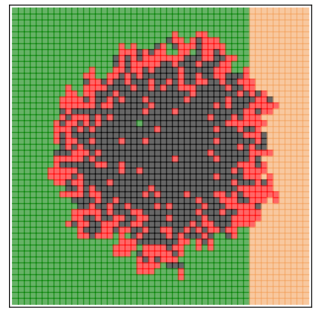

Fig. 1: A forest wildfire (red are fires, black are burnt areas) threatening a forest (green) and an urban region (light brown). We use branching processes as an approximation to predict relevant quantities, such as growth rate, and build effective constrained control policies for heterogeneous stochastic growth processes.

I. Introduction

We consider controlling a class of discrete time, discrete space, stochastic processes represented by a lattice structure. Each node on the lattice is a Markov process whose dynamics are influenced by its neighbors on the lattice. Many large-scale spatial spreading phenomena are described by this class of models, including forest wildfires, robot swarms, disease epidemics, computer viruses, and social networks [1]–[5]. Many of these processes naturally contain heterogeneous properties and we directly consider two types of heterogeneity. First, the stochastic process may propagate at different rates in different directions on the lattice. Second, each node on the lattice may have a unique discrete space, discrete time, Markov model describing its state evolution.

There is significant interest in the control of stochastic growth processes. In California, 10 of the 20 most destructive wildfires in state history have occurred within the last four years, including the current most destructive wildfire in November 2018 [6]. In addition, the number of wildfires and their intensity will increase in the US [7]. Similarly, large-scale outbreaks of infectious diseases will likely increase due to many factors, such as increased global travel and urbanization [8].

This research was supported in part by NSF grant IIS-1646921 and in part by the Cyber Valley Initiative, the IMPRS-IS, and the Max Planck Society. We are grateful for this support.

[1]Department of Mechanical Engineering, Stanford University, Stanford, USA rhaksar@stanford.edu

[2]Max Planck Institute for Intelligent Systems, Intelligent Control Systems Group, Stuttgart, Germany {solowjow,trimpe}@is.mpg.de

[3]Department of Aeronautics & Astronautics, Stanford University, Stanford, USA schwager@stanford.edu

We therefore specifically address applying control actions with a limited control resource constraint. Controlling natural phenomena is only meaningful with a resource constraint as typically the unconstrained policy is straightforward, e.g., applying fire retardant to all trees in a forest at every time interval or providing medicine to every person in a country at every time interval. While our framework is capable of describing any graph-based growth process, several quantities depend directly on the specific model formulation. Therefore, we use a heterogeneous forest wildfire process on a lattice as an example to illustrate our approach; see Fig. 1.

Our framework is predicated on using the Galton-Watson branching process as an approximation to forecast the process growth over several time intervals. Using these quantities, policies can prioritize nodes to achieve multiple control objectives. For the forest wildfire, we prioritize the preservation of urban areas over the containment of fire in the forest. We also draw connections to bond percolation from statistical physics to analyze the process stability. While there is significant literature on using lattice-based models for natural phenomena, there is limited work on the constrained control problem for very large model descriptions. In contrast to relevant prior work, we are able to model and control heterogeneous processes with resource constraints.

The main contributions of this work are: 1) We describe the map of a heterogeneous lattice-based Markov process to the bond percolation model; 2) We build a model approximation using branching processes to estimate process properties; 3) We develop novel policies that suppress a growth process under resource constraints; 4) We provide conditions for a policy to stabilize the process; 5) We show our approach significantly outperforms previous methods on a forest wildfire model with 10^{1255} states.

The remainder of this work is organized as follows. Section II reviews relevant prior work. We introduce a lattice-based Markov model of a heterogeneous forest wildfire process and then describe mapping this model to bond percolation in Section III. In Section IV we create an approximation framework based on the Galton-Watson branching process to predict relevant quantities of the stochastic growth process. In Section V we propose control policies based on the process approximation and discuss conditions under which a policy will stabilize the original process. We demonstrate our approach in comparison to prior work with simulations in Section VI and provide concluding remarks in Section VII.

II. Prior Work

The control of stochastic processes with limited resources and large state and action spaces has been studied in several areas. Frameworks have been introduced to generate policies for Markov models with large state and action spaces, including factored MDPs and graph-based MDPs (GMDPs) [9], [10]. In general, linear programming approaches are used for the unconstrained problem [11]–[13]. In previous work [14], we extended the GMDP framework to scale to large state and actions spaces while still respecting control constraints. However, this approach requires assumptions on transition distribution, which we do not require for this work. Constrained MDP formulations include explicit control constraints and several approximate methods have been proposed [15]–[19]. However, these methods are not suitable for large models due to their computational complexity. In contrast, our approach satisfies a resource limit and scales easily to large models by introducing a model approximation.

Stochastic processes on a lattice can be viewed as a process on a complex network and control methods have been proposed in this field; a recent survey is [20]. However, these methods are most appropriate for continuous-time dynamical systems models and many do not scale to large models. We instead focus on discrete time, discrete space, models to develop a tractable approach.

Percolation models have been studied extensively but have been used almost exclusively for modeling phenomena without control, such as in physics, materials science, and others [21]–[23]. Notable exceptions are [24], [25], which first proposed control policies within the percolation model framework. However, these works are limited to the homogeneous case which greatly simplifies the model and policy analysis. We significantly extend this work by directly addressing heterogeneous processes and we allow for general randomized and deterministic policy descriptions. In addition, while [25] develops a model approximation to analyze the stability of a policy, we create a different approximation that allows us to predict quantities of the stochastic process.

TABLE I: Tree transition probabilities for wildfire model. Blank entries are zero.

		x_i^{t+1}		
		H	F	B
x_i^t	H	$(1-\alpha_i)^{f_i^t}$	$1-(1-\alpha_i)^{f_i^t}$	
	F		$\beta_i - \Delta\beta_i a_i^t$	$1-\beta_i+\Delta\beta_i a_i^t$
	B			1

III. Modeling

We describe a lattice-based Markov model of a heterogeneous forest wildfire process and then discuss mapping this model to bond percolation on the square lattice.

A. Lattice-based Markov Model

The forest is modeled as a finite 2D lattice. The state of lattice node i at time t is represented by x_i^t and the set $N(i)$ refers to the lattice neighbors of node i. The state of all nodes on the lattice at time t is represented by $x^t = \{x_i^t \; \forall i\}$. There are two types of nodes, a *tree* and an *urban area*. The dynamics of both types are represented by a discrete space, discrete time, Markov model. A tree has three possible states,

$$x_i^t \in \{\text{Healthy}, \text{On Fire}, \text{Burnt}\} = \{H, F, B\}.$$

A healthy tree remains healthy unless at least one neighboring node is on fire, in which case the parameter $0 \leq \alpha_i \leq 1$ determines the likelihood that the tree will be on fire at the next time interval. This transition is based on the number of neighboring trees or urban areas on fire,

$$f_i^t = \sum_{j \in N(i)} \mathbf{1}_F(x_j^t),$$

where the notation $\mathbf{1}_A(y)$ represents the indicator function which is one when $y \in A$ and zero otherwise. A tree on fire remains on fire with likelihood determined by the parameter $0 \leq \beta_i \leq 1$. Lastly, a burnt tree remains burnt for all time.

Control actions at each time interval are binary, $a_i^t \in \{0, 1\}$, and reflect the choice of whether or not to treat a tree or urban area. For a tree on fire, choosing to treat a tree decreases the likelihood that it will remain on fire by $\Delta\beta_i$ where $0 \leq \Delta\beta_i \leq \beta_i$. We assume that choosing to treat a healthy tree or a burnt tree has no effect. Table I summarizes the dynamics of trees. An urban area has four possible states,

$$x_i^t \in \{\text{Healthy}, \text{On Fire}, \text{Burnt}, \text{Removed}\} = \{H, F, B, R\}.$$

The dynamics for an urban area are similar to the tree dynamics and the values α_i and β_i parameterize the transition probabilities; the main difference is the control of healthy urban areas. Treating a healthy urban area *removes* it from the lattice: it is no longer capable of catching on fire or spreading fire to other trees or urban areas. This represents a controlled burn or structure removal that firefighters use to limit the spread of a wildfire. In addition, treating an urban area on fire increases the likelihood it burns out. Table II summarizes the dynamics of urban areas. By choosing non-uniform values of α_i and β_i across the lattice, the wildfire will propagate at different rates in different directions. Next, we define a control model for the Markov process.

TABLE II: Urban area transition probabilities for wildfire model. Blank entries are zero.

		x_i^{t+1} H	x_i^{t+1} F	x_i^{t+1} B	x_i^{t+1} R
x_i^t	H	$(1-a_i^t)(1-\alpha_i)^{f_i^t}$	$(1-a_i^t)(1-(1-\alpha_i)^{f_i^t})$		a_i^t
	F		$\beta_i - \Delta\beta_i a_i^t$	$1-\beta_i+\Delta\beta_i a_i^t$	
	B			1	
	R				1

B. Control Model

We assume there are a limited number C of robotic agents available to apply control at each time interval t. A control policy assigns these agents to at most C unique nodes on the lattice to apply treatment. For trees or urban areas on fire, the result is a reduction of β_i by $\Delta\beta_i = \Delta\beta \ \forall i$, which is independent of the node type (tree or urban area) and identity i. The agents move faster than the spread of the wildfire and thus are able to apply treatment instantly.

C. Heterogeneous Bond Percolation

We now introduce the bond percolation model to analyze the lattice-based Markov model behavior as a function of the parameters α_i and β_i and the control policy. The 2D bond percolation model consists of nodes on an infinite square lattice where a bond exists between two nodes with a probability independent of other nodes [21]; this probability is called the *bond percolation parameter*. There exists a critical value for this parameter above which there is a path of connected nodes of infinite length [26], [27].

The forest wildfire, and other stochastic growth processes, are considered bond percolation with persistence [2]. For a wildfire, the "persistence" nature is due to nodes on fire being able to spread fire over multiple time intervals until it transitions to burnt or there are no healthy neighbors.

The bond percolation parameter for two nodes i and j is denoted by p_{ij}. For the wildfire process, consider two neighboring nodes where node i is on fire and node $j \in N(i)$ is healthy; the node type does not modify the following derivation. In the absence of control, $a_i^t = 0 \ \forall i$, the probability that node i never causes node j to transition to on fire is,

$$\begin{aligned} 1 - p_{ij} &= \sum_{t=1}^{\infty} \beta_i^{t-1}(1-\beta_i)(1-\alpha_j)^t \\ &= \frac{(1-\beta_i)(1-\alpha_j)}{1-\beta_i(1-\alpha_j)}, \end{aligned}$$

based on the dynamics in Tables I and II. After algebraic manipulation,

$$p_{ij}(\alpha_j, \beta_i) = \frac{\alpha_j}{1-(1-\alpha_j)\beta_i}. \tag{1}$$

In general, $p_{ij} \neq p_{ji}$, due to the potential uniqueness of α_i and β_i for different nodes. This property is a significant difference from the homogeneous case where $p_{ij} = p_{ji} = p \ \forall i, j$. We build a model approximation that directly addresses the uniqueness of the percolation parameter, which we present in Section IV.

Given a lattice-based model, the parameter p_{ij} is computed for all pairs of nodes and represents the likelihood of fire continuing to spread if the node i were to catch on fire. The following theorem provides conditions for two types of process behavior for percolation models.

Theorem 1 (Theorems 3.1, 3.2 in [27])**.** *Let $G = (V, E)$ be a countably infinite connected graph, with vertex set V and edge set E, that represents the square lattice. Let $p' = \{p'_e \in [0,1] \mid e \in E\}$ be the set of percolation parameters where one parameter is associated with each edge.*

(i) *If,*
$$\forall e \in E \quad p'_e \leq \frac{1}{2}, \tag{2}$$
there exists almost surely (a.s.) no infinite cluster.

(ii) *If there exists $\delta > 0$ such that,*
$$\forall e \in E \quad p'_e \geq \frac{1}{2} + \delta, \tag{3}$$
then there exists a.s. exactly one infinite cluster.

The parameter p'_e, which is associated with a graph edge, is equivalent to the parameter p_{ij}, which is computed for a pair of lattice nodes by (1). For the wildfire process, a "cluster" is a subset of nodes that are either on fire or burnt and a path exists between any two nodes. Therefore, we refer to a process as *subcritical* if the condition (2) is met as the majority of trees and urban areas are not expected to be eventually burnt. Conversely, a process is *supercritical* if the condition (3) is met and the majority of trees and urban areas are expected to be eventually burnt.

While Theorem 1 is valid for infinite square lattices, very large finite lattices exhibit similar behavior. We also include finite lattice effects in our model approximation. Given Theorem 1, the following theorem relates model parameters to a supercritical wildfire.

Theorem 2 (Critical Parameters for Percolation)**.** *If there exists a $\delta > 0$ such that,*

$$\forall i, j \qquad \frac{\alpha_j}{1-(1-\alpha_j)\beta_i} \geq \frac{1}{2} + \delta, \tag{4}$$

then the forest wildfire process is supercritical.

Proof. For a pair of nodes on the lattice the equivalent percolation parameter is determined by (1). The statement then follows from (3) in Theorem 1. □

The Markov model defined in the previous section describes the probabilistic state evolution of individual nodes. The percolation model, in contrast, specifies a "spreading" probability between nodes and then characterizes the resulting process. Theorem 2 provides a bridge between these two modeling perspectives.

Thus far, we have discussed the properties of the lattice-based Markov model in the absence of control. In the context of classical linear feedback control, the open-loop process dynamics are unstable without control when the condition in (3) is met. Feedback control is then used to produce a closed-loop system which is stable. Control actions reduce the percolation parameter p_{ij} by modifying the dynamics of the Markov model and if the condition in (2) is met then the system is stable. Therefore, one possible control objective is to stabilize a heterogeneous growth process, which we discuss further in Section V.

Based on Tables I and II, there are two cases to consider for the influence of control actions, depending on whether fire may spread to a healthy tree or a healthy urban area. If node i is on fire (either a tree or an urban area) and its neighbor node j is a healthy tree, then the change in percolation parameter is computed using (1) as,

$$\Delta p_{ij,1}(a_i^t, a_j^t) = p_{ij}(\alpha_j, \beta_i) - p_{ij}(\alpha_j, \beta_i - \Delta\beta_i a_i^t),$$

since a healthy tree is not influenced by control. If node i is on fire and its neighbor node j is a healthy urban area,

$$\Delta p_{ij,2}(a_i^t, a_j^t) = p_{ij}(\alpha_j, \beta_i) - p_{ij}(\alpha_j(1 - a_j^t), \beta_i - \Delta\beta_i a_i^t),$$

since urban areas can be removed from the lattice. The effect of control in the percolation model is thus,

$$\Delta p_{ij}(a_i^t, a_j^t) = \begin{cases} \Delta p_{ij,1}(a_i^t, a_j^t) & \text{if } x_i^t = F, x_j^t = H, j \text{ is a tree} \\ \Delta p_{ij,2}(a_i^t, a_j^t) & \text{if } x_i^t = F, x_j^t = H, j \text{ is an urban area} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

for the wildfire process. Lastly, we note that at each time interval there is a limited subset of lattice nodes that contribute to the continued spread of the process. For the forest wildfire, these are the nodes that are on fire and have at least one healthy neighbor.

Definition 1 (Growth Boundary)**.** A node on the lattice is part of the *growth boundary* $\mathcal{B}^t$ at interval t if it is on fire, $x_i^t = F$, and at least one neighbor is healthy,

$$h_i^t = \sum_{j \in N(i)} \mathbf{1}_H(x_j^t) > 0. \tag{6}$$

In the next section, we introduce the Galton-Watson branching process to predict the expected future size and stopping time for a stochastic growth process on a finite lattice.

IV. Galton-Watson Branching Process Model

The Galton-Watson branching process has been used to model population dynamics and is defined on a directed acyclic graph [28]. The graph starts with a single (root) node that produces a limited number of *children* with a prescribed probability distribution. These children become the next *parents* that can produce another group of children. The graph is organized by *generations*, which includes a set of parents, the number of children they can produce, and the probability distribution for producing children.

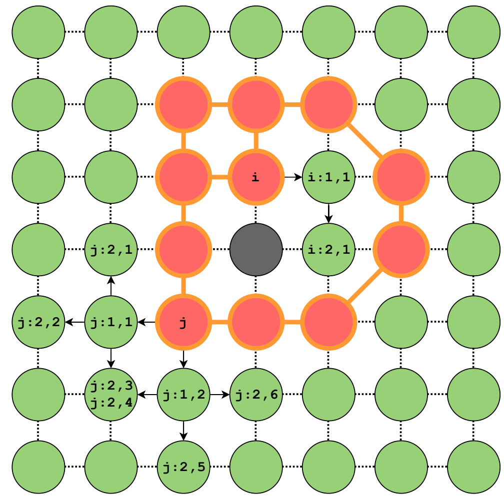

Fig. 2: Example lattice state x^t for the wildfire process where green represents healthy trees, red are on fire, and black are burnt. The orange line indicates the boundary $\mathcal{B}^t$. The branching process model is illustrated for nodes i and j where arrows indicate the possible growth of the process in one and two generations, which is analogous to a prediction of one and two future time intervals. Other node labels contain the boundary node, generation number, and a unique identifier, e.g., `j:1,2` refers to the second node that node j could spread to in one generation. One node has two labels due to the branching process model; see Fig. 3.

For the wildfire process, we use a branching process to represent the spreading dynamics of each node in the boundary $\mathcal{B}^t$ at each time interval. The first children are healthy nodes that may transition to on fire due to the boundary node. These children then become the next set of parents that may further spread fire to additional healthy nodes. Therefore, the parent nodes in each generation represent the nodes that may form part of the boundary at future time intervals.

Percolation on a lattice is not a branching process as there are multiple paths between any two nodes on the lattice. To compute the likelihood of the true process spreading further, it is necessary to enumerate a combinatorial number of paths on the lattice which is not feasible. Therefore, we instead assume each boundary node is a branching process that ignores how other boundary nodes may spread.

Let $\mathcal{GW}_i$ be a heterogeneous Galton-Watson (GW) branching process associated with boundary node $i \in \mathcal{B}^t$ at interval t; see Fig. 2. Each generation n of the $\mathcal{GW}_i$ process has an associated (potentially unique) children distribution $Y_{i,n}$. The quantity $Z_{i,n}$ describes the expected size of the n^{th} generation with $Z_{i,0} = 1$. We refer to the collection of processes as $\mathcal{GW}_{\mathcal{B}^t} = \{\mathcal{GW}_i \mid i \in \mathcal{B}^t\}$. The benefit of using GW processes is the simplicity in computing statistics of each generation [29]. The probability that process $\mathcal{GW}_i$

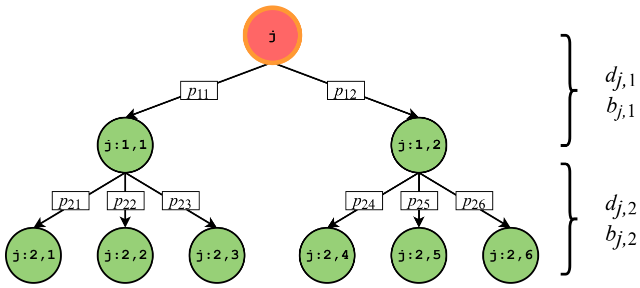

Fig. 3: Equivalent branching process representation for boundary node j in Fig. 2. The node with two labels in Fig. 2 is considered two unique nodes (here, `j:2,3` and `j:2,4`) for the branching model to avoid cycles in the graph. Note that p_{23} may not equal p_{24} due to the different parent nodes as indicated by (1). Without control, $d_{j,1} = \frac{1}{2}\sum_{k=1}^{2} p_{1k}$ and $b_{j,1} = 2$ for generation one and $d_{j,2} = \frac{1}{6}\sum_{k=1}^{6} p_{2k}$ and $b_{j,2} = 3$ for generation two.

stops at generation n is,

$$s_{i,n} = \mathbb{P}[Z_{i,n} = 0] = g_{i,1}(g_{i,2}(\cdots g_{i,n}(0))),$$

where $g_{i,j}$ is the probability generating function (PGF),

$$g_{i,n}(x) = \sum_{k=0}^{\infty} \mathbb{P}[Y_{i,n} = k]x^k.$$

The expected size of each generation is,

$$\mathbb{E}[Z_{i,n}] = \prod_{j=1}^{n} \mathbb{E}[Y_{i,j}].$$

To predict the process growth using GW processes, we must specify the possible children for each parent and the probability distribution for producing children within each generation. In the wildfire model, this corresponds to determining how fire may spread given a lattice state x^t and how likely fire will propagate to different healthy nodes. We first define a function to specify the children of a parent.

Definition 2 (Node Children)**.** For the wildfire model with lattice state x^t, the children of a node are the neighbor nodes that are healthy,

$$\mathcal{C}^t(i) = \{j \mid j \in N(i) \text{ and } x_j^t = H\}.$$

When determining the children of parent nodes, nodes that are contained in a previous generation as parents are ignored to prevent cycles in the process. For the wildfire process, we use the binomial distribution to specify the the likelihood of fire spreading and so,

$$\begin{aligned} g_{i,n}(x) &= (1 + (x-1)d_{i,n})^{b_{i,n}} \\ \mathbb{E}[Y_{i,n}] &= b_{i,n} d_{i,n} \end{aligned}$$

where we call $d_{i,n}$ the *child rate* and $b_{i,n}$ the *branching factor*. The child rate is based on the percolation parameter to capture the heterogeneous nature of the process,

$$d_{i,n} = \frac{\sum_{r \in \mathcal{P}_{n-1}} \sum_{c \in \mathcal{C}^t(r)} p_{rc} - \Delta p_{rc}(a_r^t, a_c^t)}{\sum_{r \in \mathcal{P}_{n-1}} |\mathcal{C}^t(r)|}, \tag{7}$$

Algorithm 1 Branching Process Model

1: **Input:** Lattice state x^t, control policy
2: **Output:** Predicted boundary size and stopping time
3: Determine growth boundary $\mathcal{B}^t$
4: Associate a $\mathcal{GW}$ process with each node in $\mathcal{B}^t$
5: **for** $j = 1, \ldots, n$ generations **do**
6:     **for** each process $\mathcal{GW}_i, i \in \mathcal{B}^t$ **do**
7:         Determine children of generation j
8:         Calculate child rate $d_{i,j}$ with policy (7)
9:         Calculate branching factor $b_{i,j}$ (8)
10:         Add unique children as parents of next generation
11:     Compute stopping time $s_{\mathcal{B}^t,j}$ (9)
12:     Compute expected boundary size $\mathbb{E}[Z_{\mathcal{B}^t,j}]$ (10)

where $\mathcal{P}_{n-1}$ refers to the set of parent nodes of generation $n-1$. The branching factor is then the ratio,

$$b_{i,n} = \frac{\sum_{r \in \mathcal{P}_{n-1}} |\mathcal{C}^t(r)|}{|\mathcal{P}_{n-1}|}. \tag{8}$$

Finally, for the set of GW processes $\mathcal{GW}_{\mathcal{B}^t}$,

$$s_{\mathcal{B}^t,n} = \prod_{i \in \mathcal{B}^t} s_{i,n} \tag{9}$$

$$\mathbb{E}[Z_{\mathcal{B}^t,n}] = \sum_{i \in \mathcal{B}^t} \mathbb{E}[Z_{i,n}] \tag{10}$$

are the stopping probability and expected number of nodes on fire at generation n, respectively. Predicting over multiple generations using branching processes corresponds to predicting the process growth over multiple time intervals.

Fig. 3 shows the GW process approximation of a boundary node for the example lattice state in Fig. 2. The root node, which is also the parent of generation one, is the node on fire itself. The children of generation one are `j:1,1` and `j:1,2` and each may have a unique percolation parameter. The number of children and the percolation parameters are used to determine the child rate (7) and branching factor (8). The same process repeats for the second generation with the children of generation one now serving as the parents. In this example, there is also a shared child that is double-counted in generation two which is necessary to prevent graph cycles.

Algorithm 1 summarizes the use of the GW process approximation for each time interval. Since the boundary $\mathcal{B}^t$ is not known exactly after the first generation, the estimated quantities (9) and (10) are used with the policy (line 8). For example, the UBT policy (Section V) uses the known quantity $|\mathcal{B}^t|$ for the the first generation and then the generated estimate $\mathbb{E}[Z_{\mathcal{B}^t,j}]$ for subsequent generations.

By predicting the process growth over multiple generations, it is possible to build more effective policies. For example, in Fig. 2, the boundary node i will stop spreading after two generations with probability one. Therefore, this node can safely be ignored and control resources should instead be used for other nodes. In the next section, we introduce control policies that satisfy a resource limit and characterize the conditions for a policy to stabilize a supercritical process.

V. Control Policies

We first define two benchmark randomized policies based on previous work. All of the following policies strictly satisfy a resource limit $C \geq 0$, so that $\sum_i a_i^t \leq C \ \forall t$.

Definition 3 (Uniform Boundary Treatment (UBT) [24])**.** Choose C fires in the boundary $\mathcal{B}^t$ with uniform probability to treat at each time interval.

Definition 4 (Degree Weighted Treatment (DWT) [14])**.** Choose C fires in the boundary $\mathcal{B}^t$ with probability,

$$\frac{h_i^t}{\sum_{i \in \mathcal{B}^t} h_i^t},$$

at each time interval, where h_i^t is the number of healthy neighbors (6). The quantity h_i^t can be interpreted as the out-degree of each boundary node.

Next, we define two novel deterministic control policies. The first policy, receding horizon treatment (RHT), uses the percolation parameters over multiple generations of a boundary node to estimate its rate of growth, which we call the "volatility" of a node on fire.

Definition 5 (Receding Horizon Treatment (RHT))**.** Rank all nodes in the boundary $\mathcal{B}^t$ in order of highest to lowest estimated volatility (Algorithm 2) where the parameter k is the number of generations to consider. Treat C nodes on fire with the highest estimated volatility at each time interval.

The second policy, Urban Safety Treatment (UST), is summarized in Algorithm 3. This policy first checks if any node on the boundary $\mathcal{B}^t$, after predicting k generations, will include an urban area. If so, the urban areas are ranked, in order of highest to lowest, by the number of boundary nodes that included a given urban area in their associated branching process. Up to C urban areas are then removed. Any remaining control is used to treat nodes on fire in the boundary. The ordering of boundary nodes includes their proximity to the right-most lattice edge (line 4) due to the arrangement of urban areas on the lattice; see Section VI. Lastly, if urban areas are removed, then nodes on fire with low volatility (line 3) are ignored. This is intended to eliminate boundary nodes that will stop spreading fire after k generations; an example of this is node i in Fig. 2.

Given a control policy, a relevant question is whether or not the policy will change a supercritical process to a subcritical one. The following theorem provides a condition for a policy to be considered stabilizing.

Theorem 3 (Stabilizing Policy)**.** *A policy is stabilizing if,*

$$\forall t \quad \forall i \in \mathcal{B}^t, j \in N(i): \quad p_{ij} - \Delta p_{ij}(a_i^t, a_j^t) \leq \frac{1}{2}. \quad (11)$$

Proof. By (2), a policy must reduce the percolation parameter below $\frac{1}{2}$ in order to change a supercritical process to a subcritical process. Consider the lattice state x^t at interval t. Any node that is not part of the boundary $\mathcal{B}^t$ cannot spread fire to a healthy node in which case $p_{ij} = 0$. Therefore, the policy must modify the percolation parameter of all nodes

Algorithm 2 Estimated Volatility

function $V_{\text{est}}(r, k)$
  if $k = 0$ **then return** $\frac{\sum_{c \in \mathcal{C}^t(r)} p_{rc}}{|\mathcal{C}^t(r)|}$
  else return $\sum_{c \in \mathcal{C}^t(r)} V_{\text{est}}(c, k-1)$

Algorithm 3 Urban Safety Treatment (UST)

1: **if** fire can reach an urban area in k generations **then**
2: Remove up to C urban areas that are reachable by at least one fire, with priority determined by the number of fires that reach a given urban area.
3: Remove fires with $V_{\text{est}} < k$ from consideration for treatment with remaining control.
4: Use remaining control to treat nodes in the boundary $\mathcal{B}^t$ with the highest ranking according to,
   1) proximity to the right-most lattice edge and,
   2) estimated volatility (Algorithm 2).

that can spread fire, which by definition are those in the boundary $\mathcal{B}^t$, to satisfy (2) at t. Thus, if the policy achieves this for all time intervals t, then (2) is always satisfied, and the statement follows. □

Analysis of policies using Theorem 3 must be done on a case by case basis as it depends on the policy description. For the UBT policy, only nodes on fire in the boundary will be treated so $a_j^t = 0 \ \forall t$ in (5). At each interval t, this policy chooses boundary nodes with uniform probability without replacement and so each node has probability $C/|\mathcal{B}^t|$ of being treated. Therefore, if the following holds,

$$\forall t \ \forall i \in \mathcal{B}^t, j \in N(i): p_{ij} - \frac{C}{|\mathcal{B}^t|}\Delta p_{ij}(a_i^t = 1, a_j^t = 0) \leq \frac{1}{2},$$

then the UBT policy is stabilizing according to Theorem 3. While this condition cannot be computed a priori, it can serve as real-time feedback to indicate if the resource limit C is insufficient to stabilize the process.

For the DWT policy, there is no simple analytical description of the probability for weighted sampling without replacement [30]. For the DWT, RHT, and UST policies, (11) must be evaluated at each time interval to determine if each policy is stabilizing. We present simulation results for the previously described policies in the next section.

VI. Experiments

A. Model, Policy, and Simulation Parameters

The lattice is a square of size 50×50 nodes. The right edge is composed of 500 urban areas and the remaining nodes are trees. Each simulation is initialized with fires at the center as shown in Fig. 4. The initial set of fires was chosen so that at the first time interval, there are a number of "interior" fires which will only spread for a few time intervals before extinguishing. The parameters α_i and β_i were varied across the lattice as shown in Fig. 5 and the control effectiveness parameter was set to $\Delta\beta = 0.35$.

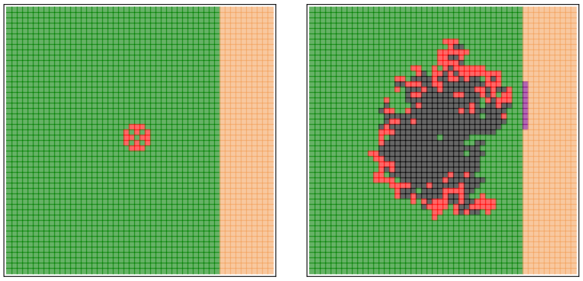

Fig. 4: (both) green nodes are healthy trees, red are on fire, black are burnt, and light brown are healthy urban areas. (left) Initial condition for simulations. (right) Example snapshot of the UST policy where purple indicates removed urban areas. By removing urban areas, the UST policy prevents other urban areas from catching on fire.

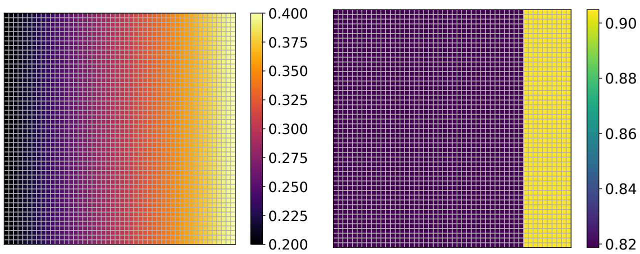

Fig. 5: (left) Values of α_i for all nodes on the lattice. (right) Values of β_i for all nodes on the lattice. These parameter values correspond to a supercritical forest wildfire.

A total of $1,000$ total simulations were run with two different resource limits, $C = 6$ and $C = 10$, for each policy. For a given policy, the fraction of remaining healthy trees and remaining healthy urban areas were recorded at the end of each simulation run. Simulations ended when there were no more nodes on fire. For the UST policy, the fraction of removed urban areas was also recorded. For the RHT policy, two horizons were tested, $k = 1$ and $k = 3$, and for the UST policy the horizon was $k = 5$.

B. Results

Tables III and IV present the median of the results and Fig. 6 provides the full distribution of the results for each policy. Without control, the process is supercritical and Tables III and IV show that the majority of the nodes are eventually burnt.

The UBT and DWT policies are not very effective with the lower resource limit $C = 6$ as they do not account for heterogeneous properties or for potential interior fires. However, with the increased capacity $C = 10$ these policies are more successful and are able to preserve more trees and urban areas. The RHT policy easily outperforms the UBT and DWT policies for both resource limits due to directly considering differences in the percolation parameter. However, this policy is only effective in preserving urban areas for the higher resource limit. In contrast, the UST policy is either equally or more effective than all other

TABLE III: Simulation results for resource limit $C = 6$ and $1,000$ simulations. The median fraction of removed urban areas for the UST policy is 3.10%.

Method	Median Remaining Healthy Trees (%)	Median Remaining Healthy Urban Areas (%)
No Control	3.80	0.00
UBT	13.25	0.02
DWT	16.32	0.20
RHT $k = 1$	22.78	1.00
$k = 3$	38.02	0.60
UST	38.78	96.90

TABLE IV: Simulation results for resource limit $C = 10$ and $1,000$ simulations. The median fraction of removed urban areas for the UST policy is 0.40%.

Method	Median Remaining Healthy Trees (%)	Median Remaining Healthy Urban Areas (%)
No Control	3.80	0.00
UBT	56.07	1.80
DWT	85.42	98.90
RHT $k = 1$	88.27	100.00
$k = 3$	90.70	100.00
UST	88.90	99.60

policies in preserving both trees and urban areas. Even for a low resource limit, UST preserves the majority of urban areas as desired. Fig. 4 shows an example snapshot of the lattice state for a single simulation while using the UST policy. The policy starts to remove urban areas once fire spreads too closely thus preventing urban areas from being burnt.

The distributions of the results, shown in Fig. 6, show that the median does not adequately capture the performance of each policy. All policies have large variance, although the RHT and UST policies significantly improve the likelihood that a majority of trees and urban areas are preserved. Only the UST policy reliably preserves urban areas for both resource limits as evidenced by the much lower variance, although this comes with the trade off of more trees being burnt and some urban areas being removed. Fig. 6 also provides a sense of how many times a given policy was stabilizing for all time intervals. Policies with large variance in the results did not frequently meet the requirements of Theorem 3 over $1,000$ simulations.

VII. Conclusions

In this work, we developed a framework for heterogeneous stochastic growth processes to predict relevant process quantities and implement general stochastic and deterministic policies with resource limits. The core of the framework is based on using branching processes, which allow for straightforward calculation of useful statistics, to approximate percolation on a lattice and generate effective control policies. We also characterized conditions for a process to be supercritical and for a control policy to stabilize the process, i.e., modify the process to the subcritical case. For future work, we plan to demonstrate our approach on

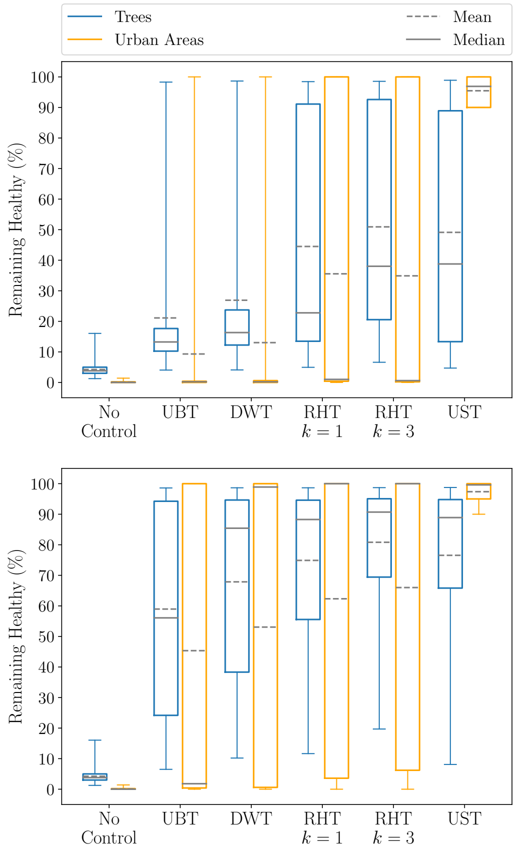

Fig. 6: Box and whisker plots of the simulation results for resource limit $C = 6$ (top) and $C = 10$ (bottom) for different policies. The whiskers represent the minimum and maximum and the box shows the first quartile, mean, median, and third quartile. Only the UST policy is capable of reliably preserving the urban areas as other policies show a large variance in their performance.

other application domains. In addition, it would be valuable to develop conditions to determine the likelihood a control policy will be stabilizing a priori. We are also investigating modifications of the agent model for scenarios where the agent travel time is not faster than the process and control cannot be arbitrarily applied anywhere on the lattice at each time interval.

References

[1] C. Moore and M. E. J. Newman, "Epidemics and percolation in small-world networks," *Phys. Rev. E*, vol. 61, pp. 5678–5682, May 2000.

[2] C. Favier, "Percolation model of fire dynamic," *Physics Letters A*, vol. 330, no. 5, pp. 396–401, 2004.

[3] J. Balthrop, S. Forrest, M. E. J. Newman, and M. M. Williamson, "Technological networks and the spread of computer viruses," *Science*, vol. 304, no. 5670, pp. 527–529, 2004.

[4] D. Easley and J. Kleinberg, *Networks, crowds, and markets*. Cambridge University Press Cambridge, 2010, vol. 8.

[5] D. Acemoglu and A. Ozdaglar, "Opinion dynamics and learning in social networks," *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, Mar 2011.

[6] CAL FIRE, "California fire historical incident statistics," http://cdfdata.fire.ca.gov/incidents/incidents_statsevents, Accessed 2019-02-22.

[7] U. G. C. R. Program, "National climate assessment report," https://nca2014.globalchange.gov/report, Accessed 2019-03-05.

[8] N. Madhav, B. Oppenheim, M. Gallivan, P. Mulembakani, E. Rubin, and N. Wolfe, "Pandemics: Risks, impacts, and mitigation," in *Disease Control Priorities: Improving Health and Reducing Poverty*, D. T. Jamison, H. Gelband, S. Horton, P. Jha, R. Laxminarayan, C. N. Mock, and R. Nugent, Eds. The World Bank, 2017, ch. 17.

[9] C. Boutilier, R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representations," *Artificial Intelligence*, vol. 121, no. 1-2, pp. 49–107, 2000.

[10] N. Forsell and R. Sabbadin, "Approximate linear-programming algorithms for graph-based Markov decision processes," in *Proceedings of the European Conference on Artificial Intelligence*, 2006, pp. 590–594.

[11] F. Chen, Q. Cheng, J. Dong, Z. Yu, G. Wang, and W. Xu, "Efficient approximate linear programming for factored MDPs," *International Journal of Approximate Reasoning*, vol. 63, pp. 101–121, 2015.

[12] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, "Efficient solution algorithms for factored MDPs," *Journal of Artificial Intelligence Research*, vol. 19, pp. 399–468, 2003.

[13] P. Robbel, F. Oliehoek, and M. Kochenderfer, "Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs," in *AAAI Conference on Artificial Intelligence*, 2016.

[14] R. N. Haksar and M. Schwager, "Controlling large, graph-based MDPs with global control capacity constraints: An approximate LP solution," in *57th IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 35–42.

[15] N. Meuleau, M. Hauskrecht, K.-E. Kim, L. Peshkin, L. P. Kaelbling, T. Dean, and C. Boutilier, "Solving very large weakly coupled Markov decision processes," in *AAAI Conference on Artificial Intelligence*, 1998, pp. 165–172.

[16] H.-J. Schütz and R. Kolisch, "Approximate dynamic programming for capacity allocation in the service industry," *European Journal of Operational Research*, vol. 218, no. 1, pp. 239 – 250, 2012.

[17] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, 2012.

[18] D. F. Ciocan and V. Farias, "Model predictive control for dynamic resource allocation," *Mathematics of Operations Research*, vol. 37, no. 3, pp. 501–525, 2012.

[19] F. Ye, H. Zhu, and E. Zhou, "Weakly coupled dynamic program: Information and Lagrangian relaxations," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 698–713, 2018.

[20] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Systems Magazine*, vol. 36, no. 1, pp. 26–46, Feb 2016.

[21] J. W. Essam, "Percolation theory," *Reports on Progress in Physics*, vol. 43, no. 7, pp. 833–912, July 1980.

[22] M. Sahini and M. Sahimi, *Applications of percolation theory*. CRC Press, 1994.

[23] S. Solomon, G. Weisbuch, L. de Arcangelis, N. Jan, and D. Stauffer, "Social percolation models," *Physica A: Statistical Mechanics and its Applications*, vol. 277, no. 1, pp. 239 – 247, 2000.

[24] A. Somanath, S. Karaman, and K. Youcef-Toumi, "Controlling stochastic growth processes on lattices: Wildfire management with robotic fire extinguishers," in *53rd IEEE Conference on Decision and Control (CDC)*, Dec 2014, pp. 1432–1437.

[25] A. Somanath and S. Karaman, "An optimal randomized policy for controlling stochastic growth processes on lattices," in *55th IEEE Conference on Decision and Control (CDC)*, Dec 2016, pp. 6240–6245.

[26] H. Kesten, "The critical probability of bond percolation on the square lattice equals $1/2$," *Communications in Mathematical Physics*, vol. 74, no. 1, pp. 41–59, 1980.

[27] G. R. Grimmett and I. Manolescu, "Bond percolation on isoradial graphs: criticality and universality," *Probability Theory and Related Fields*, vol. 159, no. 1, pp. 273–327, Jun 2014.

[28] K. B. Athreya and P. E. Ney, *Branching Processes*. Springer, Berlin, Heidelberg, 1972.

[29] D. Assaf, L. Goldstein, and E. Samuel-Cahn, "An unexpected connection between branching processes and optimal stopping," *Journal of applied probability*, vol. 37, no. 3, pp. 613–626, 2000.

[30] A. Fog, "Calculation methods for Wallenius' noncentral hypergeometric distribution," *Communications in Statistics: Simulation and Computation*, vol. 37, no. 2, pp. 258–273, 2008.