The Combinatorics of Coordinate Percolation

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Mathematics

by

Elizabeth Moseman

DARTMOUTH COLLEGE

Hanover, New Hampshire

May 31st, 2007

Examining Committee:

(chairman) Peter Winkler

Peter Doyle

Dan Rockmore

Mike Albertson

Charles K. Barlowe, Ph.D.
Dean of Graduate Students

Copyright by

Elizabeth Moseman

2007

Abstract

In coordinate percolation on an positive orthant of the n-dimensional lattice $\mathbb{Z}^n$ with a given threshold value, $1+t$, random variables $a_i(k)$ are uniform on $[0,1]$ to each pair of integers (i,k) with $1 \leq i \leq n$ and $k \geq 0$. A point $v = (v_1, \ldots, v_n)$ is open if $a_1(v_1) + a_2(v_2) + \cdots + a_n(v_n) < 1 + t$ and closed otherwise. What is the probability, $\theta(t)$, that the origin is in an infinite open cluster? Through the examination of several equivalent processes in the case where $n = 2$, an explicit expression is obtained for this probability:

$$\theta(t) = \begin{cases} 0 & t \leq 0 \\ \frac{1+3\varphi}{5} t^{2-\varphi} + \frac{4-3\varphi}{5} t^{1+\varphi} & 0 < t < 1 \\ 1 & t \geq 1 \end{cases}$$

where φ is the golden ratio, $(1+\sqrt{5})/2$.

In addition, we present some initial results toward the more general situation, where $n > 2$. In particular, we have obtained one algorithm to determine whether any given choice of $a_i(j)$ for all $1 \leq i \leq n$ and $k \geq 0$ has an infinite open path containing the origin.

Acknowledgements

This thesis is dedicated to my fourth grade teacher, Rita Fleming, without whose early encouragement and support I would have found the world of academics too lonely to bear. Thank you for a year where nothing was restricted and all avenues were explored.

I would also like to thank Art Benjamin, who made the world of discrete math enjoyable and approachable, which influenced my eventual choice of discipline, as well as the faculty and students for the Fall 2000 Budapest Semester in Mathematics, the faculty for opening my eyes to a much wider world of discrete mathematics, and the students, especially my roommates Stephanie and Brittany, for reminding me that there was more in Budapest than math.

I am grateful to Shariar Shariari and Mike Orrison for guiding my senior research. The remainder of the math faculty at Harvey Mudd helped me by making the math department a fun place to be.

In my graduate career, I required a reminder about how fun combinatorics could be, and needed to learn that there is actually research in this area, and David Little and Rosa Orellana helped me in this endeavor. Ken Bogart deserves a much more

prominent place in these acknowledgements then I am able to give him, being more than a mentor. I am certainly missing your advice, and this document would be better with your input.

Peter Winkler is the guiding force behind my thesis. I took some shoving, and thank you for doing it. Graham Brightwell made significant comments and suggestions on an early draft, resulting in a much more readable document. I would also like to thank Mike Albertson, Peter Doyle and Dan Rockmore for their input.

Finally, for emotional support, I couldn't be here without my husband, John, and a multitude of friends. My mother is to be thanked for encouraging me to follow my dreams, my sister Vivian for her help with the mathematics, and my sister Valerie for giving me an example.

Thank you, everyone.

Contents

Abstract **ii**

Acknowledgements **iii**

1 Introduction **1**

1.1 Independent Percolation 4

1.2 The Critical Phenomenon 6

1.3 Coordinate Percolation 8

1.4 This Document 9

2 Coordinate Percolation **11**

2.1 Definitions 11

2.2 Problem Statement 13

2.3 Properties 18

3 Algorithmic Approach **23**

3.1 One Dimensional "Percolation" 24

3.2 Backtracking in Percolation 27

3.3 An Algorithm for Percolation 30
3.4 Percolation at the Critical Threshold 37
3.5 An Integral Equation 39

4 Combinatorial Approach **48**
4.1 Limiting the Problem 49
4.2 Analyticity 55
4.3 Computing $P(m, n)$ 58
4.4 A Closer Look 63

5 Worms **65**
5.1 Definitions 66
5.2 The Existence of Worms 71
5.3 The Sum Worm 76
5.4 The Set of Worms Forms a Lattice 81
5.5 Counting Worms 84
5.6 Worms and Percolation 87

6 A Combined Approach **89**
6.1 Associating Instances and Permutations 90
6.2 A Stochastic Process 95
6.3 The Percolation Probability 99

7 Generalizations **103**
7.1 Lattice Generalizations 104

7.2 Distribution Generalizations 110

7.3 Operation Generalizations 112

Notation List **113**

Index **115**

Bibliography **117**

List of Tables

3.1 Algorithm for percolation 46

3.2 Algorithm for percolation at $t = 0$ 47

4.1 A correspondence table for $g(t)$. 53

4.2 Values of $P(m, n)$ 59

6.1 A process with probability of success $\theta(t)$. 94

6.2 A process by which we can build random permutations, and decide whether they percolate. 97

7.1 Possible orders when $n = 3$ 106

7.2 States for a 3-dimensional percolation algorithm 107

List of Figures

1.1 The main question 1

1.2 Collision-based coordinate percolation. 3

1.3 Comparison-based coordinate percolation 4

2.1 Example coordinate percolation configurations 15

2.2 Example coordinate percolation configurations 16

2.3 A forbidden configuration 20

3.1 Reducing backtracking 29

3.2 Examples of χ_t 43

3.3 More examples of χ_t 45

4.1 Configurations so that $\tilde{f}(m,n,\sigma) \neq \tilde{f}(m,n,\tau)$. 51

4.2 One Case in the proof of Theorem 4.3.1 60

4.3 Another case in the proof of Theorem 4.3.1 61

4.4 Another case in the proof of Theorem 4.3.1 63

4.5 Examples of the correspondence p_t when $m = n = 2$ 64

5.1 Example of a schedule 66

5.2 Example of a transpose schedule 67
5.3 Example of a witness for $\preceq$ 68
5.4 Another example witness 68
5.5 A witness of $\preceq$ for concatenation 70
5.6 Witnesses for Theorem 5.2.1, case 1 72
5.7 Witnesses for Theorem 5.2.1, case 4 73
5.8 Witnesses for Theorem 5.2.1, case 5 74
5.9 Construction of the schedule φ' 78
5.10 A witness for Theorem 5.3.1, case 1. 80

7.1 A configuration where backtracking may be required. 104
7.2 Coordinate percolation on the triangular lattice 109

Chapter 1

Introduction

What is percolation?

Consider a pumice stone (or some other porous medium) immersed in water (or another fluid). In the pumice stone, there are pockets that the water would like to fill, but there are also walls that block the flow of water. Does the water reach the center of the stone?

We can model a pumice stone with a graph where the pockets are vertices and

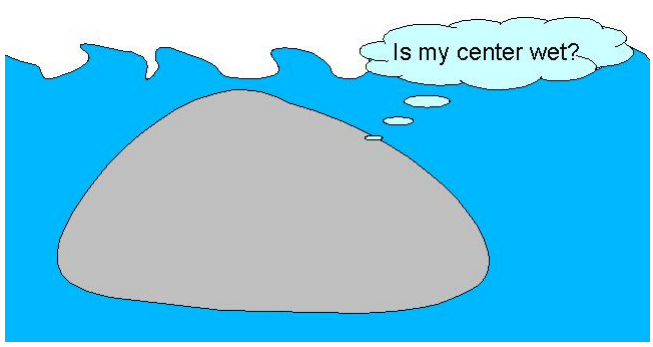


Figure 1.1: An illustration of the main question of percolation.

edges exist between adjacent pockets. This assumes that all pumice stones have basically the same structure, the only difference being whether or not the edges are blocked by a wall. In a specific stone, we also decide whether each edge is blocked (closed) or available to water (open). For the center of the stone to get wet when immersed in water, there must be a path containing from the vertex at the center of the graph to a vertex on the outside edge of the stone that only uses open edges. Since it is usually the case that the size of the stone is large compared to the size of the pockets, the model frequently model the stone with an infinite graph and determines whether there is an infinite open path beginning at the designated center vertex.

This is the basic setup for a percolation model. It includes a graph and a probability function on the edge set of the graph to determine which edges are open and which are closed. We may also place a probability function on the vertex set of the graph; in this case, an open path is a path containing only open vertices. When the probability function is on the edge set, this is called **bond percolation**; when it is on the vertex set, it is called **site percolation**.

The most frequent model of percolation uses the infinite integer lattice, in two or more dimensions. This graph has vertices at every ordered n-tuple of integers, and edges where the n-tuples differ in only one coordinate, and the differing coordinate differs by exactly 1. One model on this graph seen most frequently is **independent site percolation** (or **independent bond percolation**, when the probability function is on the edges), where each vertex is open independently with probability p, for a fixed p, with $0 < p < 1$. In this Introduction, we will consider independent

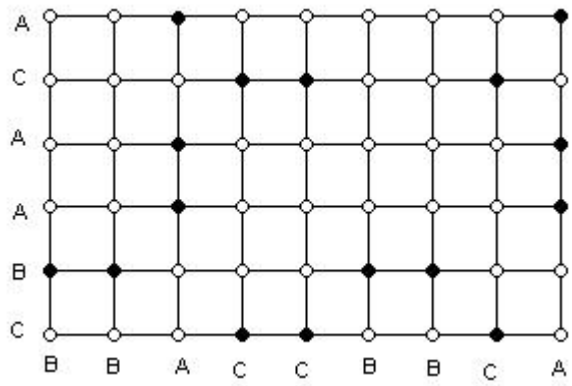


Figure 1.2: **Collision-based coordinate percolation:** Black vertices are closed and white vertices are open. Axes are labelled with the discrete random variable associated to that coordinate, chosen uniformly at random from $\{A, B, C\}$.

percolation, focusing on what is known and what is unknown, so we can contrast it with the model studied in the remainder of the document, a form of site percolation called **coordinate percolation**.

The idea of coordinate percolation is that vertices are open and closed based on random variables associated with the coordinates of the vertex, introducing a dependency between vertices. One form of coordinate percolation, introduced in [5] and studied in [1], [4], [6], [7], [8], and [10], has identical discrete random variables associated to each coordinate. A vertex is open in this model if the variables in different coordinates don't match (see Figure 1.2).

In this document, we focus on comparison-based coordinate percolation, where each coordinate receives a continuous random variable and a vertex is open if the sum of the variables in each coordinate do not exceed a given **threshold** value. For an example, see Figure 1.3. More detail on the definitions will be given in Chapter 2.

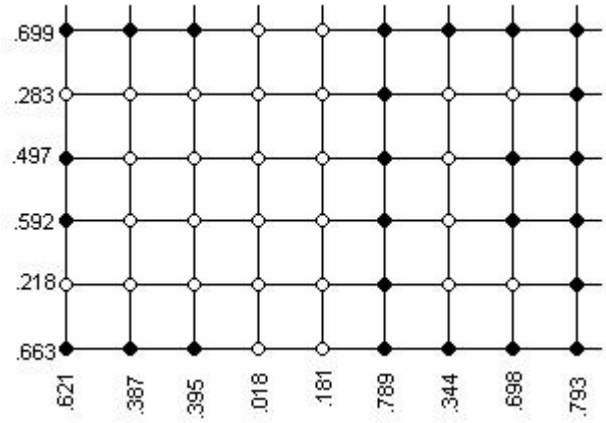


Figure 1.3: **Comparison-based coordinate percolation:** Black vertices are closed and white vertices are open. Axes are labelled with the continuous random variable associated to that coordinate, chosen uniformly at random from $[0, 1]$. The threshold value is 1.

1.1 Independent Percolation

As mentioned above, in independent site percolation each vertex is open independently with probability p where $0 < p < 1$. There are many applications of this model.

Sensor grids

To protect an estate, a very dense grid of sensors is laid down around it. Each sensor has a probability p of being out of service at any given time. What is the probability that an intruder can get across the grid?

City Networks

To create a network in New York, fiber optic cable is laid down along all the streets. Each cable has the same number of different time slots which may be used. Each of the time slots on each cable is filled with probability $1 - p$. To transmit a message between two places, there must be a single slot that is open at every edge along a path between the places. What is the probability that such a path can be found?

Chemistry

In a phase transition, molecules form bonds with nearby molecules with a probability p, where p depends on the temperature. (A lower temperature corresponds to a higher probability.) How large a structure will be formed at a given temperature?

Storm Drains

A large grid of pipes is laid out to catch and contain rainwater. Unfortunately, silt and debris tend to catch in the intersections of the pipes, so each intersection has a probability $1 - p$ of being blocked. How much water can be held in this system?

To answer the question of sensor grids, the network is modelled as independent site percolation, and we want to find the probability, $\theta(p)$, that there is an open path between the estate and the edge of the sensor grid. For the city network, if there were only one time slot this would be a straightforward application of independent bond percolation, so the probability of a single slot being open would be a different probability function $\theta_{bond}(p)$. Since there are multiple slots available, we can view this as n simultaneous bond percolation processes, so the probability of one open

slot would be $1 - (1 - \theta(p))^n$.

The chemistry and storm drain applications depend on a different value, called the **cluster size**. The cluster at a vertex v is the set of vertices which may be reached by open paths from that vertex. The function $\chi(p) = E[|C|]$ is the expected size of the cluster C, where C contains the distinguished vertex. The chemistry example uses independent bond percolation while the storm drains example, as phrased, uses independent site percolation. The function $\chi(p)$ would be different in each of these cases.

A more realistic model of the storm drain system would allow both the intersections and the pipes to be blocked, making a mixed model for percolation. Since the functions $\theta(p)$ and $\chi(p)$ are unknown in both site and bond percolation, attempting to work with this model may be ambitious.

1.2 The Critical Phenomenon

Consider the sensor grid example. If the probability of failure, p, of any given sensor is small, then it is extremely likely that there will not be a path of failed sensors across the grid. However, if the probability of failure is high, most of the sensors will be out and an intruder only has to avoid these few in order to make it across. In fact, most forms of percolation exhibit what is called a **critical phenomenon** or **phase transition**. This means that there is a probability p_c, called the **critical probability**, so that $\theta(p) = 0$ if $p < p_c$ and $\theta(p) > 0$ if $p > p_c$. All of the examples in Section 1.1 use the two dimensional lattice $\mathbb{Z}^2$, but it is easy to see why percolation

may use other graphs, notably the three dimensional lattice $\mathbb{Z}^3$. Mathematicians have generalized to the finite dimensional lattice $\mathbb{Z}^n$, but much is still unknown regarding most of the questions in percolation for any n. For example, the critical probability, p_c, is $\frac{1}{2}$ for independent bond percolation on $\mathbb{Z}^2$, but is unknown for higher dimensions and the critical probability is unknown for independent site percolation in all dimensions, although bounds are known for these values [9].

Since the critical probability is crucial to the understanding of percolation, some effort has been made to understand the percolation probability, $\theta(t)$ and the cluster size $\chi(p)$ for p near p_c. One question which is not yet completely answered is the value of $\theta(p_c)$. Other than at p_c, $\theta(p)$ is known to be a continuous function, and it is known to be continuous at p_c when $n = 2$ and $n \geq 19$, but the question is still open in the remaining dimensions. To understand the behavior near p_c, we introduce the critical exponents

$$\begin{aligned}
\gamma &= -\lim_{p \to p_c^-} \frac{\ln \chi(p)}{\ln |p - p_c|}, \\
\beta &= \lim_{p \to p_c^+} \frac{\ln \theta(p)}{\ln |p - p_c|}, \text{ and} \\
\delta^{-1} &= -\lim_{n \to \infty} \frac{\ln \Pr(|C| \geq n)}{\ln n}.
\end{aligned}$$

In general, it is not known whether or not these exist. We interpret the critical exponents as the limiting functions of the expected cluster size, percolation probability, and actual cluster size. In other words, as $p \to p_c^-$, the expected cluster size $\chi(p)$ looks like $(p_c - p)^{-\gamma}$. As $p \to p_c^+$, the percolation probability $\theta(t)$ looks like $(p - p_c)^{\beta}$.

At p_c, the probability that the cluster containing the origin has n vertices looks like $n^{-1/\delta}$. In independent percolation, the values of all of these critical exponents are unknown.

1.3 Coordinate Percolation

The remainder of this document focuses on coordinate percolation. Coordinate percolation arises in scheduling problems, as seen in [2], but the following application to resource allotment is another motivation for studying this problem.

A computer needs to run n different programs. For $n = 3$, these programs might be something like the operating system, the web browser, and matlab. Each of these programs has an infinite list of upgrades. Unfortunately, each of these upgrades must be installed consecutively. Although the amount of system resources each upgrade uses is random, it is known in advance. Can all of the upgrades be installed without going over the amount of resources the computer has?

In this model, the amount of resources each upgrade uses is a random variable with uniform distribution on the interval $[0, 1]$. This is unrealistic. However, the approach generalizes to random variables with uniform distribution on any interval, as long as the size of the interval is the same for all programs. This generalization is explored in Chapter 7. In the case where we have a random variable with uniform distribution on the interval $[0, 1]$, this can be thought of as percentage of resources of an average system. Each system is different, so the total amount of system resources varies and is given as the threshold. The threshold can be thought of as the amount of

resources available in comparison to the average system. If the amount of resources all of the programs together are trying to use exceeds the amount of resources available on the system, the system will not run.

This is a percolation process because we may define a vertex (i, j, k) to be open if the total resources used by the ith upgrade of the operating system, the jth upgrade of the web browser and the kth upgrade of matlab does not cause the system to seize.

The probability of being able to install all the upgrades may be changed if we are allowed to downgrade, returning to a usage of system resources which has already been passed; this possibility is explored in Section 2.3. It will certainly change with the system (the threshold) and with the number of programs to be run (the dimension). One thing to notice with this model is that if the amount of resources on the system isn't as much as that on the average system, there is no chance of being able to perform all the upgrades, since at least one of them will use more resources than this system has. However, it is unclear from this problem if the average system may perform all the upgrades, or even if a system with less than n times the resources of the average system may perform all the upgrades.

1.4 This Document

Throughout the remainder of the document, the percolation model will be coordinate percolation on the n-dimensional integer lattice unless otherwise specified. Chapter 2 focuses on the problem set-up, and discusses definitions. We also discuss the properties of this model, including the existence of a critical threshold and a

forbidden configuration. Chapter 3 outlines an algorithmic approach to computing the percolation probability, $\theta(t)$. This leads to an algorithm to answer the decision question, "Given a configuration, does it percolate?" in any dimension. The algorithm is analyzed and adapted in the case of $n = 2$ to yield an integral equation. Chapter 4 outlines another approach, which seeks to reduce the configuration space to something finite. We use the existence of random variables with value less than t to truncate the infinite sequences of random variables. This approach yields an expression for $\theta(t)$ in terms of a combinatorial object known as a worm. We explore worms and their relationship to coordinate percolation in Chapter 5. Chapter 6 uses an approach which combines aspects of algorithms and a reduction in the size of the sample space. This approach uses another adaptation of the algorithm introduced in Chapter 3, but in this adaptation only the order of the random variables is observed. This leads to an explicit formula for the probability $\theta(t)$, which is shown in Section 6.3. We conclude, in Chapter 7, by exploring various generalizations.s

Chapter 2

Coordinate Percolation

In this Chapter the formal language describing coordinate percolation is developed. Section 2.1 begins the definitions necessary for the remainder of the document. Section 2.2 formally defines what a configuration is and what it means for such a configuration to percolate, as well as setting up the probabilistic background for the problem. Section 2.3 lays out some immediate properties of this percolation model.

2.1 Definitions

The basis of percolation is the notion of some vertices in a graph being **open** and the rest of the vertices being **closed**. In *independent percolation*, whether a vertex is open or closed does not depend upon the state of any of the remaining vertices. In *coordinate percolation*, the state of a vertex depends on variables associated with the coordinates of that vertex. While independent percolation can be modeled on any infinite graph, coordinate percolation requires some notion of coordinates. This

document focuses on coordinate percolation in the the nonnegative quadrant of the integer lattice $\mathbb{Z}^2$, which I will denote by $\mathbb{N}^2$. There will be some excursions into higher dimensional integer lattices. Formally, define the graph $\mathbb{N}^2 = (V(\mathbb{N}^2), E(\mathbb{N}^2))$ where

$$\begin{aligned} V(\mathbb{N}^2) &= \{(i,j) | i \geq 0, j \geq 0 \text{ are integers}\} \\ E(\mathbb{N}^2) &= \{\{(i,j),(i+1,j)\},\{(i,j),(i,j+1)\} | (i,j) \in V(\mathbb{N}^2)\} \end{aligned}$$

For the general graph $\mathbb{N}^n = (V(\mathbb{N}^n), E(\mathbb{N}^n))$ we let

$$\begin{aligned} V(\mathbb{N}^n) &= \{(i_1, i_2, \ldots, i_n) \mid i_j \geq 0, 1 \leq j \leq n \text{ are integers}\} \\ E(\mathbb{N}^n) &= \{ \{(i_1, i_2, \ldots, i_n), (i_1, \ldots, i_{j-1}, i_j+1, i_{j+1}, \ldots, i_n)\} \\ &\qquad \mid (i_1, i_2, \ldots, i_n) \in V(\mathbb{N}^n), 1 \leq j \leq n\}. \end{aligned}$$

We will denote the origin $(0, 0, \ldots, 0)$ in $\mathbb{N}^n$ as 0_n.

A **path** φ in $\mathbb{N}^n$ is a sequence $\varphi(0), \varphi(1), \ldots, \varphi(k)$ where $\varphi(i) \in V(\mathbb{N}^2)$ for $0 \leq i \leq k$, $\varphi(i) \neq \varphi(j)$ whenever $i \neq j$ and $\{\varphi(i), \varphi(i+1)\} \in E(\mathbb{N}^n)$ for $0 \leq i < k$. We say the **length** $l(\varphi)$ of the path φ is equal to k. In other words, the length of the path is one less than the length of the corresponding sequence of vertices. It is sometimes useful to look at the coordinates of the vertices in a path. We define

functions $\varphi_j : \{0, 1, \ldots, l(\varphi)\} \to \mathbb{Z}$ for $1 \leq j \leq n$ by

$$\varphi_j(i) = x_j \Leftrightarrow \varphi(i) = (x_1, x_2, \ldots, x_j, \ldots, x_n).$$

For a vertex $v \in V(\mathbb{N}^n)$ we will sometimes abuse notation and say $v \in \mathbb{N}^n$ or use $\mathbb{N}^n$ to denote the edge set of this graph.

If we already have a definition of open and closed vertices, an **open path** is a path containing only open vertices. An **open cluster** is a maximal collection of open vertices connected by open paths. An **infinite open cluster** is an open cluster containing a (countably) infinite number of vertices.

2.2 Problem Statement

This document focuses on one kind of coordinate percolation. We have a collection of independent random variables $a_i(j)$ for each integer pair with $1 \leq i \leq n$ and $0 \leq j$. The distribution for each random variable is uniform on the interval $[0, 1]$. We define the sample space Ω_n as all possible sets of values of these random variables. In other words, it is an ordered n-tuple of sequences $a_1, a_2, \ldots, a_n \colon \mathbb{N} \to [0, 1]$. In a given instance $\sigma \in \Omega_n$, we define the value of a vertex $v = (v_1, v_2, \ldots, v_n) \in \mathbb{N}^n$ as the sum $\mathrm{val}_\sigma(v) = a_1(v_1) + a_2(v_2) + \cdots + a_n(v_n)$. We are given one parameter t and define the **threshold** to be $1+t$. A vertex v is **open** if $\mathrm{val}(v) < 1+t$ and **closed** otherwise. We also define a configuration space $\mathcal{F}_n = \{\sigma \colon \mathbb{N}^n \to \{0, 1\}\}$ where each configuration specifies only which vertices v of $\mathbb{N}^n$ are open ($\sigma(v) = 1$) and which are closed ($\sigma(v) = 0$). For each n, t we also define a map $f_t \colon \Omega_n \to \mathcal{F}_n$ where $f_t(\sigma)$

is the configuration defined by the values of the vertices given by σ, i.e.,

$$f_t(\sigma)(v) = 1 \Leftrightarrow \mathrm{val}_\sigma(v) < 1 + t.$$

For each t, the allowable configurations are in the set $f_t(\Omega_n) \subset \mathcal{F}_n$. In $\mathbb{N}^2$, we denote $a_i = a_1(i)$ and $b_i = a_2(i)$ to eliminate unnecessary clutter. Figures 2.1 and 2.2 have sample configurations in the case $n = 2$ and $t = 0$. The filled squares represent closed points. Notice that there is a great deal of structure in these configurations; many of the lines are mostly open (mostly white) or mostly closed (mostly black). Mostly closed lines have a high value while mostly open lines have a low value.

Given a uniformly random $\sigma \in \Omega_n$, is 0_n is in an infinite open cluster in the configuration $f_t(\sigma)$? If the answer to this question is yes, we say $f_t(\sigma)$ **percolates**. Otherwise, the configuration **does not percolate**. To define the **percolation probability**, $\theta(t)$, we define a percolation function $\Theta_t \colon \Omega_n \to \{0, 1\}$ by

$$\Theta_t(\sigma) = \begin{cases} 0 & \text{if } f_t(\sigma) \text{ does not percolate} \\ 1 & \text{if } f_t(\sigma) \text{ percolates.} \end{cases}$$

Then $\theta(t)$ is defined as

$$\theta(t) = \Pr[\Theta(\sigma) = 1]$$

where $\sigma \in \Omega_n$ is uniformly random. It is also useful to know the probability that a uniformly random $\sigma \in \Omega_n$ is mapped to a given $\tau \in \mathcal{F}_n$. To this end we define $\rho_t \colon \mathcal{F}_n \to [0, 1]$ as

$$\rho_t(\sigma) = \Pr[f_t(\tau) = \sigma]$$

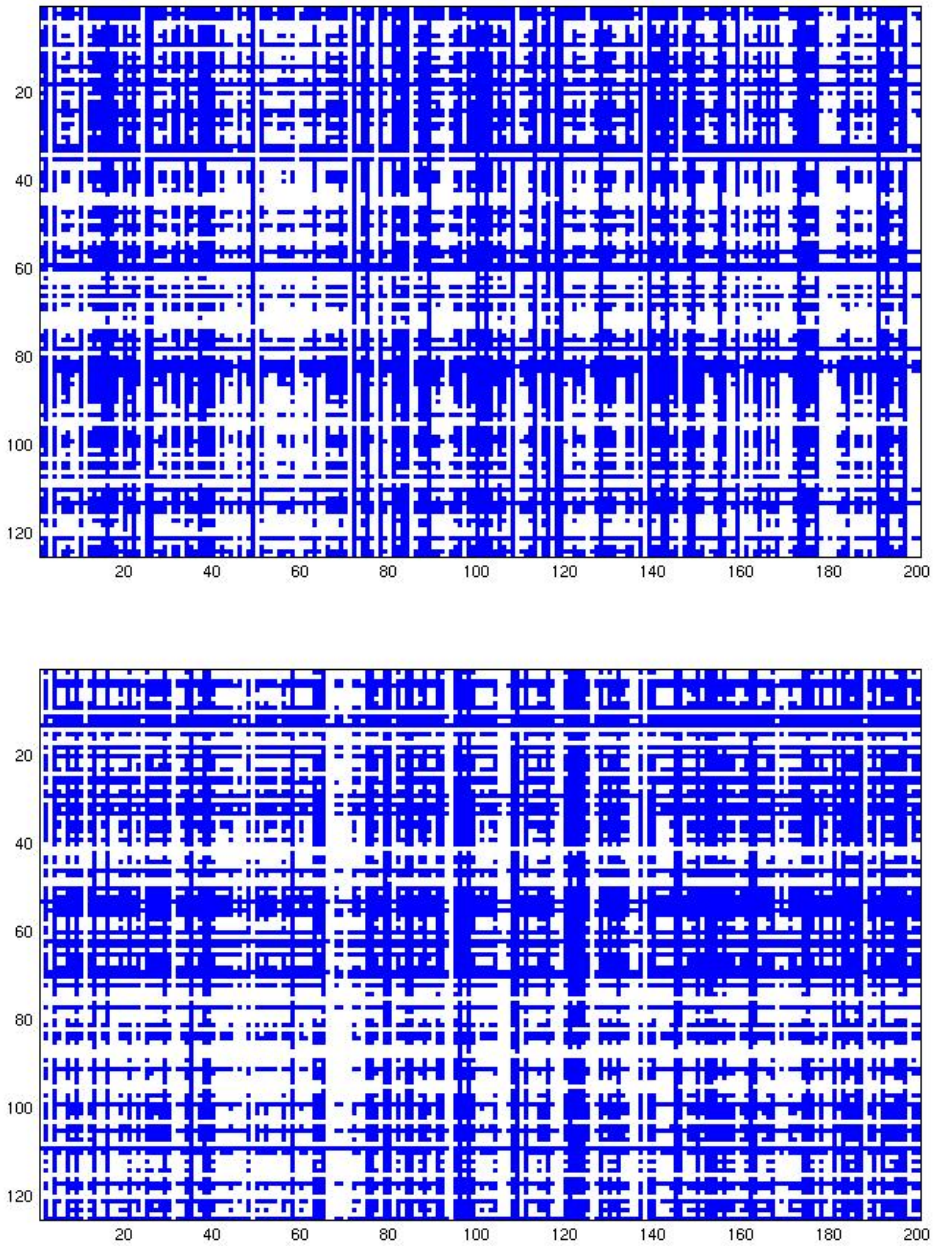


Figure 2.1: Example coordinate percolation configurations

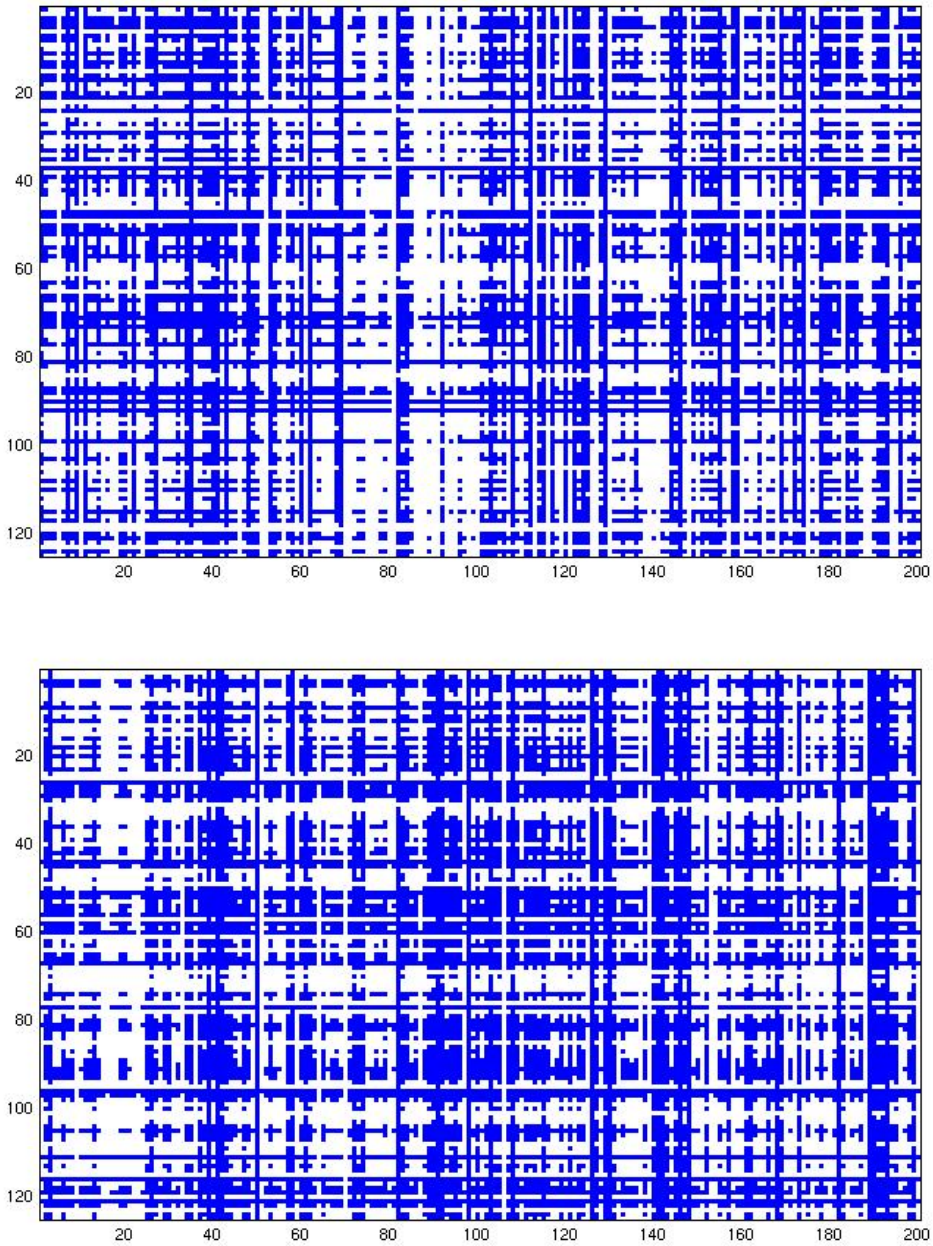


Figure 2.2: Example coordinate percolation configurations

for a uniformly random $\tau \in \Omega_n$.

Although the main focus of this work is on computing the percolation probability, it is often useful to look at **cluster size** as well. Define $C(\sigma)$ to be the open cluster containing the origin for a given configuration $\sigma \in \mathcal{F}_n$. Then the **expected cluster size**, $\chi(t)$, is defined as

$$\sum_{k=0}^{\infty} k \cdot \Pr[|C(\sigma)| = k] = \sum_{\sigma \in \mathcal{F}_n} |C(\sigma)| \rho_t(\sigma).$$

The critical exponents for coordinate percolation are the same as the critical exponents for independent percolation, namely

$$\begin{aligned} \gamma &= -\lim_{p \to p_c^-} \frac{\ln \chi(p)}{\ln |p - p_c|}, \\ \beta &= \lim_{p \to p_c^+} \frac{\ln \theta(p)}{\ln |p - p_c|}, \text{ and} \\ \delta^{-1} &= -\lim_{k \to \infty} \frac{\ln \Pr(|C| \geq k)}{\ln k}. \end{aligned}$$

The percolation probability, expected cluster size, and critical exponents all depend on the dimension n, which will always be clear from context.

We may also consider $\mathbb{N}^n$ as a digraph where each edge is oriented in the direction of the increasing coordinate. In other words

$$\begin{aligned} E(\mathbb{N}^n) &= \{ \ ((i_1, i_2, \ldots, i_n), (i_1, \ldots, i_{j-1}, i_j + 1, i_{j+1}, \ldots, i_n)) \\ &\qquad \mid (i_1, i_2, \ldots, i_n) \in V(\mathbb{N}^n), 1 \leq j \leq n \}. \end{aligned}$$

In this case, an "open cluster" is not well-defined in the previous definition. We define the **open cluster at a point** v as the set of vertices which may be reached by open paths beginning at v. In this case, known as **oriented percolation**, a configuration percolates if the open cluster at 0_n is infinite.

2.3 Properties

We begin with some basic results.

Theorem 2.3.1. *For $t < 0$, $\theta(t) = 0$. For $t > 0$, $\theta(t) > 0$. For $t > 0$, $\chi(t) = \infty$.*

Proof. To prove the first part, let $s = 1+t$ where $t < 0$. Let $\sigma = (a_1, a_2, \ldots, a_n) \in \Omega_n$ and consider the configuration $f_t(\sigma)$. With probability 1 there exists j_i for each $1 \leq i \leq n$ so that $a_i(j_i) > s$. Then any point $v = (v_1, v_2, v_3, \ldots, v_n)$ with $v_i = j_i$ for any i is closed in $f_t(\sigma)$. These points form a closed, finite box containing the origin. Since there are only a finite number of vertices inside the box, any infinite path containing the origin must contain at least one of these points. Thus none of these paths are open, and the configuration does not percolate.

For the second part, let $t > 0$ and $\sigma = (a_1, a_2, \ldots, a_n) \in \Omega_n$. With probability t/n, $a_i(0) < t/n$ for a given i. Therefore, with probability $(t/n)^{n-1}$, $a_i(0) < t/n$ for each $1 \leq i < n$. In this case, every point $(0, 0, 0, \ldots, 0, j)$ in $f_t(\sigma)$ has value $a_1(0) + a_2(0) + a_3(0) + \cdots + a_{n-1}(0) + a_n(j) < (n-1)t/n + 1 < 1 + t$ so these points are open. We call such a collection of open points an **open line**. Thus, $\theta(t) > (t/n)^{n-1} > 0$, as desired.

Finally, since $\chi(t)$ is the expected value of the cluster size and $t > 0$, we know $\theta(t) > 0$ and $\chi(t) > \theta(t) * \infty = \infty$. $\square$

One consequence of this theorem is that the **critical threshold**, t_c, the analogue in coordinate percolation of the critical probability in independent percolation, is zero. That is to say, that $\theta(t) = 0$ if $t < t_c$ and $\theta(t) > 0$ if $t > t_c$. One consequence of this theorem is that the **critical threshold**, t_c, the analogue in coordinate percolation of the critical probability in independent percolation, is zero. That is to say, that $\theta(t) = 0$ if $t < t_c$ and $\theta(t) > 0$ if $t > t_c$.

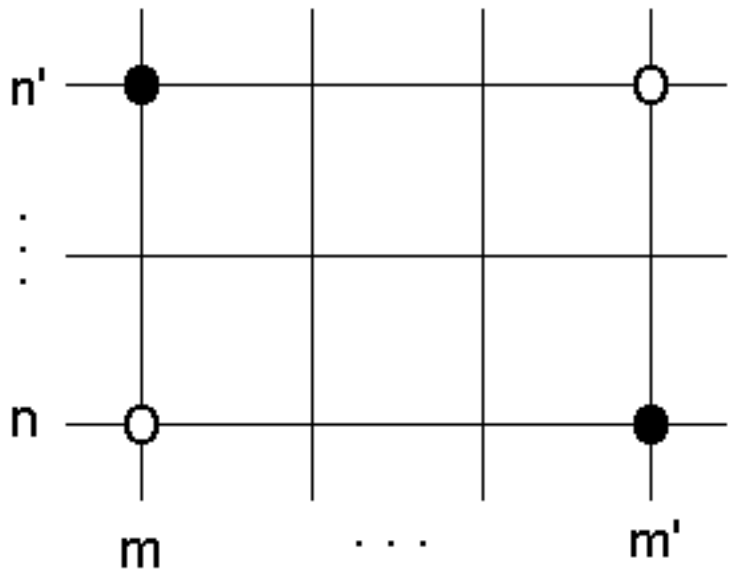
Another immediate consequence of the definition of open and closed points in coordinate percolation is a forbidden configuration.

Theorem 2.3.2. *Given a configuration $\sigma \in f_t(\Omega_2)$, for any integers m, m', n, n', if the points (m, n) and (m', n') are open then at least one of the points (m, n') and (m', n) is also open. Equivalently, if (m, n) and (m', n') are both closed then at least one of the points (m, n') and (m', n) is also closed.*

Proof. Let $\sigma \in f(\Omega_2)$. Suppose (m, n) and (m', n') are open and (m, n') is closed in σ. Let $\tau = (a, b) \in f_t^{-1}(\sigma)$. By definition, $a_m + b_n < 1+t$, $a'_m + b'_n < 1+t$, and $a_m + b'_n > 1+t$.

$$\begin{aligned} a'_m + b_n &= a'_m + (b'_n - b'_n) + (-a_m + a_m) + b_n \\ &= (a'_m + b'_n) - (b'_n + a_m) + (a_m + b_n) \\ &< (1+t) - (1+t) + (1+t) \\ &= 1+t \end{aligned}$$

Figure 2.3: **Forbidden Configuration:** The points (m, n) and (m', n') are open and the points (m, n') and (m', n) are closed. This configuration cannot happen in coordinate percolation.



So (m', n) is open, concluding the proof. $\square$

Theorem 2.3.2 gives a configuration of points which is forbidden. This is illustrated in Figure 2.3, where the filled vertices are closed and the unfilled vertices are open. This square cannot occur anywhere in an instance of the problem, no matter how many lines separate the vertices (horizontally or vertically). The absence of any forbidden configuration allows the use of diagonal moves in the grid (steps from (m, n) to $(m \pm 1, n \pm 1)$) without any change in the size or shape of the open clusters.

This forbidden configuration generalizes to higher dimensions. The proof is inductive, with the base case $n = 2$ provided by Theorem 2.3.2.

Theorem 2.3.3. *If* $(i_1, i_2, \ldots, i_n)$ *and* $(i_1 \pm 1, i_2 \pm 1, \ldots, i_n \pm 1)$ *in* $\mathbb{N}^n$ *are both open*

in a configuration $\sigma \in f_t(\Omega_n)$, then there is an open path of length n between these vertices.

Proof. Let $\sigma \in f_t(\Omega_n)$ be a configuration with open vertices of this form and $\tau = (a_1, a_2, \ldots, a_n) \in f_t^{-1}(\sigma)$. Without loss of generality, suppose the two open vertices are $0_n = (0, 0, \ldots, 0)$ and $1_n = (1, 1, \ldots, 1)$. Since they are open, $\mathrm{val}_\tau(0_n) < 1+t$ and $\mathrm{val}_\tau(1_n) < 1+t$. Consider the n vertices adjacent to 0_n, $v_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is in the ith coordinate. The average value of these vertices is

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \mathrm{val}_\tau(v_i) &= \frac{1}{n}\sum_{i=1}^{n} a_1(0) + \cdots + a_{i-1}(0) + a_i(1) + \cdots + a_n(0) \\
&= \frac{1}{n}\Big((n-1)a_1(0) + (n-1)a_2(0) + \cdots + (n-1)a_n(0) \\
&\qquad + a_1(1) + a_2(1) + \cdots + a_n(1)\Big) \\
&= \frac{1}{n}\left((n-1)\mathrm{val}_\tau(0_n) + \mathrm{val}_\tau(1_n)\right) \\
&< \frac{1}{n}\left((n-1)(1+t) + (1+t)\right) \\
&= 1+t
\end{aligned}$$

Since the average value is at most $1+t$, there must be some v_k with $\mathrm{val}_\tau(v_k) < 1+t$. This v_k is open. We now look for an open path of length $n-1$ between v_k and 1_n. Reduce to a problem of finding an open path for a configuration $\sigma' \in \Omega_{n-1}(t - a_k(1))$ as follows: Define $\tau' = ({a_1}', {a_2}', \ldots, {a_n}') \in \Omega_{n-1}$ so that

$${a_i}'(j) = \begin{cases} a_i(j) & i < k \\ a_{i+1}(j) & i \geq k \end{cases}$$

and $\sigma' = f_{t-a_k(1)}(\tau')$ and find an open path between 0_{n-1} and 1_{n-1} in the configuration σ'. With these definitions, every vertex in v with 1 as the kth coordinate is open in σ if and only if the vertex with the kth coordinate removed is open in σ'. By induction, we can find an open path φ of length $n-1$ with $\varphi(0) = 1_{n-1}$ and $\varphi(n-1) = 0_{n-1}$. Define φ' by

$$\varphi'_i(j) = \begin{cases} \varphi_i(j) & \text{if } i < k \\ \varphi_{i-1}(j) & \text{if } i > k \\ 1 & i = k \end{cases}$$

when $0 \leq j < n$ and $\varphi'(n) = 0_n$. By the construction of σ' and φ', φ' is an open path from 1_n to 0_n, proving the theorem.

$\square$

Chapter 3

Algorithmic Approach

The main idea of an algorithmic approach to percolation problems is to develop an algorithm to find an infinite path containing the origin. This approach is demonstrated in a sample problem in Section 3.1. In independent percolation, each vertex needs to be considered individually in order to find the probability of percolation, making such an algorithm impractical. This is not true in coordinate percolation since there are dependencies between vertices. Specifically, the forbidden configuration introduced in Theorem 2.3.2 and illustrated in Figure 2.3 allows us to take a path with no backtracking. This is explored in Section 3.2. With the notion of backtracking in mind, we define an algorithm to determine if a given configuration percolates in Section 3.3 and use a variation of this algorithm in Section 3.4 to show that the probability of percolation with $t = 0$ is 0. Finally, in Section 3.5 we use an algorithm to develop an integral equation describing the probability of percolation in the 2-dimensional case.

3.1 One Dimensional "Percolation"

We begin our study with a sample problem which will have some relevance later. This actually models coordinate percolation in the (probability 0) case where the instance used is $(a, a) \in \Omega_2$. For each $i \geq 0$, we assign a random variable X_i, which is uniform on the interval $[0, 1]$. In other words, the sample space is

$$P = \{(X_0, X_1, X_2, \ldots) |\ X_i \in [0, 1]\}$$

with the uniform distribution. We also have a configuration space

$$P(t) = \{\sigma \colon \mathbb{N} \to \{0, 1\}\}$$

and a function $\omega_t \colon P \to P(t)$ defined for $x = (x_0, x_1, x_2, \ldots) \in P$ so that

$$\omega_t(x)(n) = 1 \Leftrightarrow x_i + \min_{j \leq i}\{x_j\} < 1 + t.$$

We ask if there are any closed points. We define $\theta(t)$ to be the probability that $\omega_t(x)(n) = 1$ for all $n \in \mathbb{N}$ for a uniformly random $x \in P$.

To solve this problem, we use the same approach we would like to use on the main problem. First, we analyze the easy cases, then we develop an algorithm to determine if a given instance $x \in P$ percolates. This algorithm gives rise to an integral equation for the probability function, which we solve to find a closed expression for our function.

First, the easy cases. If $t < 0$, then for any $x = (x_0, x_1, x_2, \ldots) \in P$, with

probability 1 there is at least one $x_i > 1 + t$ and thus $\omega_t(x)(i) = 0$, so $\theta(t) = 0$. Notice also that if $x_0 > (1 + t)/2$, $\omega_t(x)(0) = 0$ and the instance does not percolate. At any i, let $s = \min_{j \leq i}\{x_j\}$. Clearly, if $s < t$, all the points beyond i are open in $\omega_t(x)$. Otherwise, we look for a new minimum value by iterating this process while incrementing i. Let $s' = x_j$ where $x_j \notin [s, 1 + t - s]$ and $x_k \in [s, 1 + t - s]$ for every k between i and j. There are three cases,

1. $s' < t$,
2. $t < s' < s$, and
3. $1 + t - s < s' < 1$.

In the first case, all future points are open. In the second case, we have to iterate this process with the new minimum value s' because there is still the possibility of closed points. In the third case, we have found a closed point in $\omega_t(x)$. The probabilities are as follows: the first case occurs with probability $t/(2s - t)$, the second case with probability $(s - t)/(2s - t)$ and the third case with probability $(s - t)/(2s - t)$. Since our result depends on the current value, s, we write our integral equation in terms of a supplemental function $\chi_t(s)$, the probability that the remaining points are open given that the current minimum is s. From above, we see that

$$\chi_t(s) = \frac{t}{2s - t} + \frac{s - t}{2s - t}\hat{\chi}_t(s)$$

where $\hat{\chi}_t(s)$ is the probability of winning on the next step. The value of this is the

expected value of $\chi_t(s')$ for $t < s' < s$ which is

$$\hat{\chi}_t(s) = \frac{1}{s-t}\int_t^s \chi_t(s')ds'.$$

Therefore, for $s > t$,

$$(2s-t)\chi_t(s) = t + \int_t^s \chi_t(s')ds'.$$

We solve this integral equation to get

$$\chi_t(s) = \begin{cases} 1 & s < t \\ \sqrt{\frac{t}{2s-t}} & t < s < (1+t)/2 \\ 0 s > (1+t)/2. & \end{cases}$$

The actual value of $\theta(t)$ is the integral of this supplemental function. For $0 \leq t \leq 1$,

$$\begin{aligned} \theta(t) &= 2\int_0 \chi_t(s)ds \\ &= 2\int_0^t ds + 2\int_t^{(1+t)/2} \sqrt{\frac{t}{2s-t}}ds \\ &= t + -t + \sqrt{t} \\ &= \sqrt{t}. \end{aligned}$$

Note, in particular, that $\theta(0) = 0$.

3.2 Backtracking in Percolation

What is backtracking? In general, backtracking is taking steps that don't get closer to the goal. More precisely, a path φ in the n-dimensional lattice $\mathbb{N}^n$ **backtracks in the kth coordinate** if there is an index i so that $(\varphi(i+1), \varphi(i))$ is an edge in the oriented digraph $\mathbb{N}^n$.

Theorem 3.2.1. *Given a configuration $\sigma \in f_t(\Omega_n)$, if there is an open path from $(0, 0, \ldots, 0)$ to $(m_1 - 1, m_2 - 1, \ldots, m_n - 1)$ in the finite sublattice $[m_1] \times [m_2] \times \cdots \times [m_n]$, then there is an open path with no backtracking.*

Proof. Given $\sigma \in f_t(\Omega_n)$, let φ be a shortest open path $(0, 0, \ldots, 0)$ to $(m_1 - 1, m_2 - 1, \ldots, m_n - 1)$ in the finite sublattice $[m_1] \times [m_2] \times \cdots \times [m_n]$. Let $\tau = (a_1, a_2, \ldots, a_n) \in f_t^{-1}(\sigma)$. Suppose φ backtracks in coordinate k.

Define the set $I = \{i \mid \exists j \text{ such that } \varphi_{j+1}(k) = i \text{ and } \varphi_j(k) = i+1\}$. Let $i_1 = \min I$ and $i_2 = (\max I) + 1$. Pick i_3 so that $a_k(i) > a_k(i_3)$ for any i between i_1 and i_2 (inclusive). Pick s_1, s_2 so that

$$\begin{aligned} s_1 &= \min\{s \mid \varphi_k(s) = i_3\} \\ s_2 &= \max\{s \mid \varphi_k(s) = i_3\}. \end{aligned}$$

Define a new sequence φ' as if it were a path with $\varphi'(s) = \varphi(s)$ for $0 \leq s \leq s_1$ and

$s_2 \leq s \leq l(\varphi)$, and for $s_1 < s < s_2$ let

$$\varphi'_j(s) = \begin{cases} \varphi_j(s) & j \neq k \\ i_3 & j = k. \end{cases}$$

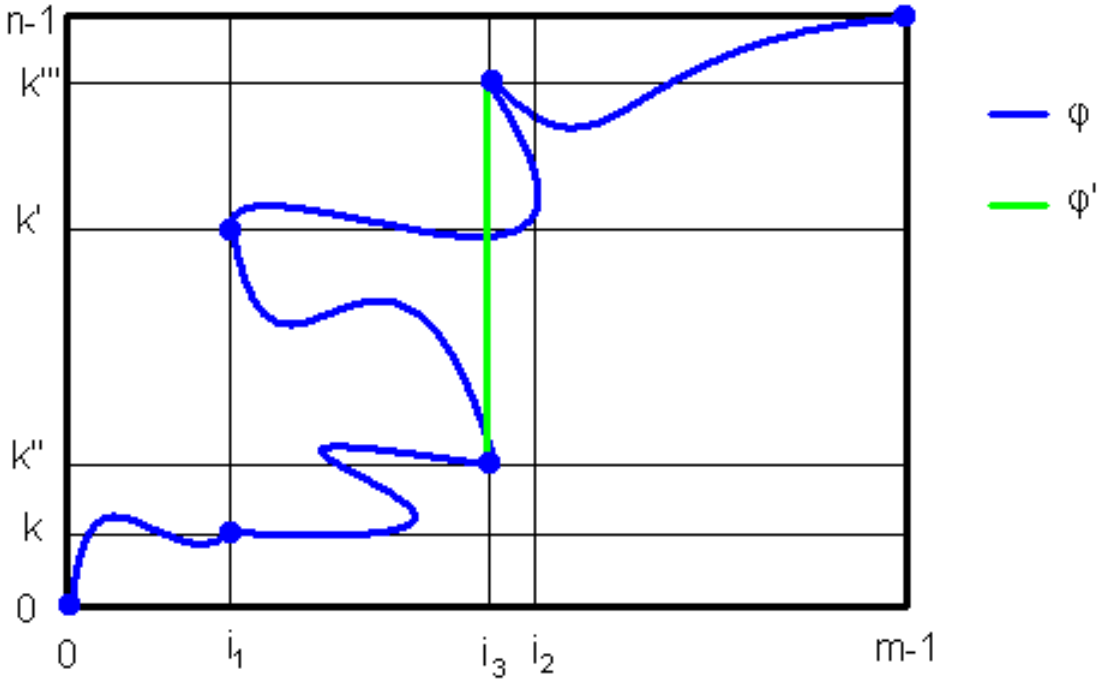
φ' between s_1 and s_2 is the projection of φ onto the n–1-dimensional subspace $x_k = i_3$ in $\mathbb{N}^n$. It is not a path since it has repeated vertices whenever φ moves in the kth coordinate. This occurs at least once since the set I is nonempty by the assumption of backtracking in this coordinate. It does define a path if the repeated vertices are removed, and this path is shorter than φ. It remains to show that all the vertices in φ' are open to arrive at a contradiction. For any s in the interval (s_1, s_2), the kth coordinate is in the interval (i_1, i_2) by the definitions of these two numbers. Thus, $a_k(i_3) < a_k(\varphi_k(s))$. Therefore,

$$\begin{aligned} \mathrm{val}\varphi'(s) &= a_1(\varphi'_1(s)) + a_2(\varphi'_2(s)) + \cdots + a_k(\varphi_k(s)) + \cdots + a_n(\varphi'_n(s)) \\ &= a_1(\varphi_1(s)) + a_2(\varphi_2(s)) + \cdots + a_k(i_3) + \cdots + a_n(\varphi_n(s)) \\ &< a_1(\varphi_1(s)) + a_2(\varphi_2(s)) + \cdots + a_k(\varphi_k(s)) + \cdots + a_n(\varphi_n(s)) \\ &= \mathrm{val}\varphi(s) \\ &< 1 + t \end{aligned}$$

so the vertex $\varphi'(s)$ is open. For s outside this interval, $\varphi(s) = \varphi'(s)$ and open by the choice of φ. This completes the proof. □

Figure 3.1 illustrates the new path in the 2-dimensional case.

Figure 3.1: In the two dimensional case, to form φ' from the black path φ as in Theorem 3.2.1, replace the middle section with the grey path.



To relate this to previous work, we need a different interpretation of the integer lattice. We wish to interpret it as a modular lattice. This is a kind of partially ordered set, so we'll need a partial order on the vertices in the integer lattice. We say a vertex (i, j) is less than or equal to a vertex (i', j') if $i \leq i'$ and $j \leq j'$ and we write $(i, j) \leq (i', j')$. A partially ordered set forms a lattice if there is a well-defined **meet** and a well-defined **join**. The **meet of** a **and** b, written $a \wedge b$, is the element c so that whenever $a \geq d$ and $b \geq d$ it is also true that $c \geq d$. Similarly, the **join of** a **and** b, $a \vee b$ is the element c so that $c \leq d$ whenever $a \leq d$ and $b \leq d$. For the integer lattice, we define the join of two elements $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ is $d = (d_1, d_2, \ldots, d_n)$ where $d_i = max\{a_i, b_i\}$ and their meet is $e = (e_1, e_2, \ldots, e_n)$

where $e_i = \min\{a_i, b_i\}$.

A **modular lattice** is a lattice which satisfies the relation $a \wedge (b \vee c) = (a \wedge b) \vee c$ for every triple of elements a, b, c with $a \geq c$. We check that the above definitions of meet and join satisfy this relation: $a \geq c$ means that $a_i \geq c_i$ for every i so if $a \wedge (b \vee c) = f$ and $(a \wedge b) \vee c = g$ then $f_i = \min\{a_i, \max\{b_i, c_i\}\}$ and $g_i = \max\{\min\{a_i, b_i\}, c_i\}$ so if $b_i \leq c_i$ then $f_i = c_i = g_i$ and if $b_i > c_i$ then $f_i = \min\{a_i, b_i\} = g_i$. A **bone** in a subset S of a modular lattice is a pair $A, B \in S$ where $A \vee B$ and $A \wedge B$ are not in S.

In the integer lattice with the subset S of open points defined by some configuration $\sigma \in \mathcal{F}_n$, a bone is one of the forbidden configurations from Theorem 2.3.2. A **maximal chain** is a path φ where $\varphi(i) < \varphi(i+1)$ and there is no element x in the lattice with the property that $\varphi(i) < x < \varphi(i+1)$ for all $0 \leq i < l(\varphi)$. These definitions show that Theorem 3.2.1 follows from the following more general theorem of Brightwell and Winkler:

Theorem 3.2.2. *[2] Let S be a boneless set in a modular lattice L. If S percolates, it contains a maximal chain.*

3.3 An Algorithm for Percolation

By Theorem 3.2.1, we only have to consider oriented coordinate percolation (which may also be defined as percolation without backtracking) in order to determine whether or not a given configuration percolates. Table 3.1 is a flowchart describing an algorithm to determine whether or not an instance $\tau \in \Omega_n$ leads to a configuration $f_t(\tau) \in \mathcal{F}_n$ with $t > 0$ which percolates.

The idea of this algorithm is to remain at the lowest possible valued vertex until a lower valued vertex is found. This is a multi-dimensional algorithm with two end states, G and H. The proof of the correctness of this algorithm will show that if the instance ends in state G then it has found a closed finite box around the origin, preventing percolation, and if it reaches the state H then it has found an open line (1-dimensional subspace) and an open path to this line, showing the configuration percolates.

The algorithm begins in State A, simply testing whether or not the origin is open in the configuration $f_t(\tau)$. State B, which is reached when the origin is open, is an initialization state. φ is the open path that this algorithm is attempting to construct. For now, it consists only of the origin. The index i represents the length of the path — $\varphi(i)$ will always be the last point that has been assigned on the path, and the smallest valued vertex which has been examined (this is shown in Lemma 3.3.1). The index k represents the current direction — the algorithm looks at further vertices in this direction. The index j_l for l between 1 and n has two functions. First, it represents how many values have been examined in direction l. At this point, the values $a_l(m)$ have been examined for every $m < j_l$ and the values $a_l(m)$ for every $m > j_l$ have not been examined. The second function of the indices j_l has to do with that remaining value, $a_l(j_l)$, which may or may not have been examined. This value indicates closed points; for $l < k$ the vertex $\mathbf{v}_l = (v_1, v_2, \ldots, v_n)$ where

$$v_m = \begin{cases} \varphi_m(i) & m \neq k \\ j_l & m = l \end{cases}$$

is closed. Since $\varphi(i)$ is the lowest valued vertex seen so far, this indicates that all points $w = (w_1, w_2, \ldots, w_n)$ with $w_m < j_m$ for $m \neq l$ and $w_l = j_l$ are closed. However, the points on the segment between $\varphi(i)$ and $\mathbf{v}_l$ are open.

Part of the function of the algorithm is to find these closed points — they close off one direction the open path may take. If n of these points are found, the configuration does not percolate, which is shown in Lemma 3.3.2. The remaining function of the algorithm is to find open lines, which is tested in State C. Lemma 3.3.3 will show that when $\mathrm{val}_\tau(\varphi(i)) - \max_l\{a_l(\varphi_l(i))\} < t$, the configuration percolates.

In State D, a new vertex v is tested to see if it is closed. If it isn't, the algorithm will test more vertices in this dimension. If it is, the algorithm asks if it has found a closed point in every dimension (State J) which would represent a closed box containing the origin (Lemma 3.3.2). If it hasn't found a closed point in every direction, it looks in the next direction by incrementing k (State I).

If the vertex v tested in State D is open, it may be a vertex with a lower value than that of $\varphi(i)$, which is tested in State E. If it isn't better, the algorithm increments j_k (State F) and tests the next vertex in the same direction. If it is better, the path φ is updated in State K to include the segment between $\varphi(i)$ and v. The index i is incremented as much as necessary so that $\varphi(i) = v$ when this state is finished. Since the path has been updated, the algorithm returns to State C where it tests if it has found an open line.

To prove the correctness of the algorithm, we establish several lemmas regarding the behavior and the values of vertices which have already been examined. This will show that if the algorithm reaches one of its terminating states, then it has reached

it correctly. Once this is established, we show that for a uniformly random τ in Ω_n, one of the terminating states is reached with probability 1.

Lemma 3.3.1. *Given $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$, at any time s, let $v = (v_1, \ldots, v_n)$ be any vertex with $v_l < j_l$ for all l. Then $\mathrm{val}_\tau(v) \geq \mathrm{val}_\tau(\varphi(i))$. In particular, $a_k(v_k) > a_k(\varphi_k(i))$ for any $v_k < j_k$.*

Proof. Let $v_k < j_k$. There is at least one $\hat{\imath}$ so that $a_k(v_k)$ was compared to $a_k(\varphi_k(\hat{\imath}))$. If it won this comparison ($a_k(v_k) < a_k(\varphi_k(\hat{\imath}))$), then $a_k(\varphi_k(i))$ would have been compared to something as small or smaller, so we may assume that it lost the comparison ($a_k(v_k) \geq a_k(\varphi_k(\hat{\imath}))$). But in order to change from $\varphi(\hat{\imath})$ to $\varphi(i)$ (with possible other transitions along the way) the values at each coordinate remain the same or are smaller, so $a_k(v_k) \geq a_k(\varphi_k(\hat{\imath})) \geq a_k(\varphi_k(i))$, as desired. □

Lemma 3.3.2. *Given $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$, if the algorithm described in Table 3.1 reaches the state G from the state J then the collection of vertices $C = \{v \mid v_l < j_l$ for all but one index l, $v_l = j_l$ for this remaining index$\}$ is a collection of closed vertices.*

Proof. Since every time a new vertex is chosen for $\varphi(i)$ the index variable k is reset to 1, we need only look at one value for this vertex. Consider any vertex $v = (v_1, v_2, \ldots, v_n)$ where $v_k = j_k$ and $v_l < j_l$ for every $l \neq k$. Since $v_l < j_l$, $a_l(v_l) \leq a_l(\varphi_l(i))$ for every $l \neq k$ by Lemma 3.3.1. Thus $\mathrm{val}_\tau(v) = a_1(v_1) + a_2(v_2) + \cdots + a_n(v_n) \geq \mathrm{val}_\tau(\varphi(i)) - a_k(\varphi_k(i)) + a_k(j_k) = \mathrm{val}_\tau(w)$ where $w = (w_1, w_2, \ldots, w_n)$ with $w_k = j_k$ and $w_l = \varphi_l(i)$ for all $l \neq k$. Notice that w is exactly the vertex used in Step D when it transitions to J, so $\mathrm{val}_\tau(w) \geq 1 + t$. Thus $\mathrm{val}_\tau(v) \geq 1 + t$ and v is closed. □

In the future, we will call such a collection of closed vertices a **closed box** containing the origin.

Lemma 3.3.3. *Given $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$, if the algorithm reaches state H, then the configuration arising from the instance τ percolates.*

Proof. Let $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$. Reaching state H means that $\mathrm{val}_\tau(\varphi(i)) - \max_l\{a_l(\varphi_l(i))\} < t$. Let k be the index of the $\max_l\{a_l(\varphi_l(i))\}$. For any vertex v with $v_l = \varphi_l(i)$ for $l \neq k$, $\mathrm{val}_\tau(v) = \mathrm{val}_\tau(\varphi(i)) - a_k(\varphi_k(i)) + a_k(v_k) < t + a_k(v_k) < t+1$. This collection of vertices is connected and infinite. Call it A. Consider $\varphi(j)$. For any j there are values i, i' where i changes to i' and $\varphi(j)$ is set during the same instance of step K. Then $j \leq i'$. If $j = i'$ then $\mathrm{val}_\tau(\varphi(j)) < \mathrm{val}\tau(\varphi(i))$ and $\varphi(i))$ is open by induction (with base case $i = 0, \varphi(0) = 0_n$). Otherwise, $j < i'$. Then there was a point v with $v_k = \varphi_k(j)$ and $\mathrm{val}_\tau(v) < 1 + t$ which was used to increment j_k after the comparison in step D. Since $a_l(\varphi_l(j)) = a_l(\varphi_l(i)) < a_l(v_l)$ for every $l \neq k$ by Lemma 3.3.1, $\mathrm{val}_\tau(\varphi(j)) < \mathrm{val}_\tau(v) < 1 + t$. Therefore the path φ is open. Thus φ and the vertices from the set A form an infinite set of connected open vertices containing the origin. Thus, there is an infinite open cluster containing the origin, as desired. □

Proof of Correctness of Algorithm. Let $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$ be uniformly random. Lemma 3.3.2 shows that if we reach state G, the instance τ does not lead to a configuration which percolates. Lemma 3.3.3 shows that if we reach state H then the instance τ does lead to a configuration which percolates. Since these are the only terminating states, it remains to show that the algorithm terminates. This is done by induction on the number of dimensions.

In two dimensions, we have two values, j_1, j_2. If j_2 does not increase, then we are stuck with the current y, $\varphi_2(i)$ and its value $a_2(\varphi_2(i))$. If $a_2(\varphi_2(i)) < t$, then the check in State C transitions to State H, and the algorithm terminates. Otherwise, with probability 1 there is a closed point on the line $y = \varphi_2(i)$. Say this point is $(x_0, \varphi_2(i))$. As j_1 increases we will either reach an open line $x = x_1$ with $x_1 < x_0$ (when $j_1 = x_1$) or we will reach the closed point (when $j_1 = x_0$), both of which cause the algorithm to terminate.

Suppose that we know the algorithm terminates in $n-1$ dimensions. We want to show that is terminates in n dimensions. In Section 3.1, we found that for a series of uniform values on $[0, 1]$, with probability 1 there is a time when the sum of the minimum and the maximum values exceeds 1. So we can pick z_0, z_1 so that $a_n(z_0) = \min_{j \leq z_1}\{a_n(j)\} < t/n$ and $a_n(z_0) + a_n(z_1) \geq 1$. Let $\tau' = ({a_1}', {a_2}', \ldots, {a_n}') \in \Omega_{n-1}$ be the instance where $a'_i(j) = a_i(j)$ for every $1 \leq i \leq n-1$ and $j \geq 0$. Run the $n-1$-dimensional algorithm with instance τ' and threshold $1 + t - a_n(z_0)$. Notice that this terminates, by the induction hypothesis.

Suppose this percolates and the final vertex is $\varphi'(i)$. Notice that the path φ' is the same as the one we would find running the algorithm on τ except that it may not terminate at the first open line because it may be the case that this depends only on the first $n-1$ dimensions. However, if it continues to run, it will find $\varphi'(i)$. Assume $j_n > z_0$ and let $z = (\varphi'_1(i), \varphi'_2(i), \ldots, \varphi'_{n-1}(i), z_0)$. Suppose $a_n(z_0) < \max_l\{a_l(\varphi'_l(i))\}$.

Then

$$\begin{aligned}\mathrm{val}_{\tau'}(\varphi'(i)) - \max\{a_n(z_0), \max_l\{a_l(\varphi'_l(i))\}\} &= \mathrm{val}_{\tau'}(\varphi'(i)) - \max_l\{a_l(\varphi'_l(i))\} \\ &< t - a_n(z_0)\end{aligned}$$

and

$$\begin{aligned}& \mathrm{val}_{\tau}(\varphi(i)) - \max\{a_n(z_0), \max_l\{a_l(\varphi'_l(i))\}\} \\ = \;& \mathrm{val}_{\tau'}(\varphi'(i)) - \max\{a_n(z_0), \max_l\{a_l(\varphi'_l(i))\}\} + a_n(z_0) \\ < \;& t - a_n(z_0) + a_n(z_0) \\ = \;& t.\end{aligned}$$

Otherwise, $a_n(z_0) > \max_l\{a_l(\varphi'_l(i))\}$ and $\mathrm{val}_\tau(z) < na_n(z_0) < n(t/n)$ by the choice of z_0. This forces φ to terminate in a point which forces the transition from step C to step H.

Otherwise, the $n-1$-dimensional algorithm with instance τ' and threshold $1+t-a_n(z_0)$ does not percolate. Suppose the final vertex is $\varphi(i)$ and the final box has sides $x_1 = j_1, s_2 = j_2, \ldots, x_{n-1} = j_{n-1}$. Then the algorithm with instance τ and threshold $1+t$ will find the box with sides $x_1 = j_1, s_2 = j_2, \ldots, x_{n-1} = j_{n-1}, x_n = z_1$, if not a smaller box. The minimum vertex in this box is $z = (\varphi'_1(i), \varphi'_2(i), \ldots, \varphi'_{n-1}(i), z_0)$. For $1 \le l \le n-1$, if we change the lth coordinate of z to j_l the value of this vertex is at least $1 + t - a_n(z_0) + a_n(z_0) > 1 + t$ by the termination of the algorithm with instance τ' and threshold $1 + t - a_n(z_0)$ in state G. Thus these vertices are

closed. The vertex $(\varphi_1'(i), \varphi_2'(i), \ldots, \varphi_{n-1}'(i), z_1)$ has value $\mathrm{val}_{\tau'}(\varphi(i)) + a_n(z_1) > t + \max_l\{a_l(\varphi_l(i))\} + a_n(z_1) > t + t/n + a_n(z_1) > t + a_n(z_0) + a_n(z_1) > 1 + t$ and is thus closed. Therefore, all of the vertices on the sides of the box are closed.

So, if $j_n > z_1$, we know the algorithm must have terminated. Otherwise, without termination, j_n has obtained its maximum. Let $\varphi(i)$ be a vertex on the path after j_n achieves its maximum. Define $\pi\colon \Omega_n \times \mathbb{N}^n \to \Omega_{n-1}$ by $(\tau, v) \mapsto \tau'$ where $a_i'(j) = a_i(j + v_i)$ for $1 \le i \le n-1$ and $0 \le j$. Notice a vertex $w' = (w_1, \ldots, w_{n-1}) \in \mathbb{N}^{n-1}$ is open in $f_{t-a_n(v_n)}(\phi(\tau, v))$ if and only if the vertex $w = (w_1 + v_1, \ldots, w_{n-1} + v_{n-1}, v_n)$ is open in $f_t(\tau)$. Run the algorithm on $\pi(\tau, \varphi(i))$ with threshold $1 + t - a_n(v_n)$. This creates a path which corresponds to the path the algorithm takes in $\mathbb{N}^n$ (the correspondence is in the direction the steps take), but we know the algorithm terminates. If it terminates in the "Percolates" state, then the n-dimensional algorithm also terminates in the "Percolates" state. If it terminates in the "Does not Percolate" state, suppose the final vertex is $\varphi'(j)$. If the vertex $z = (\varphi_1'(j) + \varphi_1(i), \varphi_1'(j) + \varphi_1(i), \ldots, \varphi_{n-1}'(j) + \varphi_{n-1}(i), j_n)$ is closed, the n-dimensional algorithm terminates in state does not percolate. Otherwise, z is open and j_n increases, contradicting the hypothesis that j_n had obtained its maximum. $\square$

3.4 Percolation at the Critical Threshold

Recall from Theorem 2.3.1 that the critical threshold is $t_c = 0$.

Theorem 3.4.1. *For $n \ge 2$, $\theta(0) = 0$.*

Proof. This proof amounts to showing that the algorithm in Table 3.2 is correct and

that it terminates with probability 1. Let $\tau = (a_1, a_2, \ldots, a_n) \in \Omega_n$. Note that at all times, $a < a_i(l)$ for every $l < j_i$ and $1 \leq i \leq n$. Reaching state J amounts to finding that $a_k(j_k) > 1-a$ for every k. Consider a vertex $w = (w_1, w_2, \ldots, w_n)$ where $w_k = j_k$ for one integer k and $w_i < j_i$ for all $i \neq k$. Thus,

$$\begin{aligned}
\mathrm{val}_\tau(w) &= a_1(w_1) + a_2(w_2) + \cdots + a_n(w_n) \\
&\geq (n-1)a + a_k(j_k) \\
&\geq (n-1)a + 1 - a \\
&> 1.
\end{aligned}$$

So all these points w are closed and form a box enclosing the origin. Thus the instance does not percolate.

To show that the algorithm terminates with probability 1, notice that we cannot leave the C $\rightarrow$ H $\rightarrow$ G $\rightarrow$ C loop until we find j_k so that $a_k(j_k) \notin [a, 1-a]$. The probability that this j_k has $a_k > 1-a$ is exactly $1/2$ when $a < 1/2$. If it isn't, the point v is closed. Since $\mathrm{val}_\tau(v) < \mathrm{val}_\tau(0_n)$ the origin is also closed, which means it is not in an open cluster. We can safely ignore this case. Thus, the probability that the algorithm does not terminate when k reaches $n+1$ is exactly $1 - 2^{-n} < 1$. In order for the algorithm to fail to terminate, every time it reaches state F this must occur. But if it does not terminate the mth time it reaches this state, then with probability 1 it will reach state F again, so it will pass through state F at least m times for every m. The probability of reaching state F $m+1$ times is $(1-2^{-n})^m$. The probability that the algorithm does not terminate is therefore $\lim_{m\to\infty}(1-2^{-n})^m = 0$. Therefore,

the algorithm terminates with probability 1 and $\theta(0) = 0$, as desired. $\square$

3.5 An Integral Equation

As we did in Section 3.1, we now want to develop an integral equation to describe $\theta(t)$. We restrict our attention to the case $n = 2$. Since the algorithm depends on the current values we are looking at, this will actually be for a supplementary function, $\chi_t(a_i, b_j)$ where the current lowest point is (i, j). In this case, $\chi_t(r, s) = 1$ if $r < t$ or $s < t$ and

$$\theta(t) = \int_0^1 \int_0^1 \chi_t(r, s) dr ds.$$

We now want to examine $\chi_t(r, s)$ for other values of r, s. Using the algorithm in Table 3.1 in the case of $n = 2$, we look at the value a_{j_1} first to find a closed point and must look at it again after finding a lower value in the other coordinate, to make sure the lower value is low enough. This causes the value to be dependent, so we adapt the algorithm slightly. We describe the entire algorithm below, and this adaptation is in step 3b. As we enter this process, we assume we have just assigned the vertex $\varphi(i) = (i_1, i_2)$ (beginning with $\varphi(0) = (0, 0)$) and recall that we will use the convention $a_1(i) = a_i, a_2(j) = b_j$. Increment i_1' starting at i_1 until one of the following occurs:

1. We find $i_1' > i_1$ with $a_{i_1'} < t$. In this case, we have reached an open line and the instance percolates.

2. We find $i_1' > i_1$ with $t < a_{i_1'} < a_{i_1}$. In this case, we iterate the process with the

new vertex (i_1', i_2), appending the segment between (i_1, i_2) and (i_1', i_2) to the path φ.

3. We find $i_1' > i_1$ with $1 + t - b_{i_2} < a_{i_1'} < 1$. The point (i_1', i_2) is closed. Now iterate i_2' starting at i_2 until one of the following occurs:

   (a) We find $i_2' > i_2$ with $b_{i_2'} < t$. In this case, we have reached an open line and the instance percolates.

   (b) We find $i_2' > i_2$ with $t < b_{i_2'} < 1 + t - a_{i_1'}$. Notice the original algorithm requires only that $b_{i_2'} < b_{i_2}$, in which case the point (i_1', i_2') may be closed. This change assures that this point is open, in which case all of the points on the segment between (i_1, i_2') and (i_1', i_2') are open. In this case, we iterate the process with the new vertex (i_1, i_2').

   (c) We find $i_2' > i_2$ with $1 + t - a_{i_1} < b_{i_2'} < 1$. The point (i_1, i_2') is closed, so we terminate the algorithm and the instance does not percolate.

Assign $r = a_{i_1}$ and $s = b_{i_2}$. As i_i' is incremented, successive values of $a_{i_1'}$ are examined until one is found in the range $R = [0, t] \cup [t, r] \cup [1 + t - s, 1]$. Let $r' = a_{i_1'}$. If these successive values are independent of the values already examined, then

$$\begin{aligned} \Pr[r' \in [0, t]] &= t, \\ \Pr[r' \in [t, r]] &= r - t, \text{ and} \\ \Pr[r' \in [1 + t - s, 1]] &= s - t, \end{aligned}$$

so $\Pr[r' \in [0,t] | r' \in R] = t/(r+s-t)$. This is the probability that the first case occurs. Similarly, the probability that the second case occurs is $(r-t)/(r+s-t)$ and the probability that the third case occurs is $(s-t)/(r+s-t)$. In case 3, we increment i_2' until we find a value $s' = b_{i_2'}$ in the range $S = [0,t] \cup [t, 1+t-r'], \cup [1+t-r, 1]$. If these successive values are independent of the values already examined, the probability of case 3a is $t/(1-r'+r)$, the probability of case 3b is $(1-r')/(1-r'+r)$, and the probability of case 3c is $(r-t)/(1-r'+r)$.

The cases are mutually exclusive events, so the probability of percolation is the sum of the probability of percolation in the case times the probability of the case occurring. Case 1 percolates with probability 1. Case 2 percolates with probability $\chi_t(r', s)$, assuming that successive iterations of the algorithm are independent. Case 3a percolates with probability 1. Case 3b percolates with probability $\chi_t(r, s')$, assuming that successive iterations of the algorithm are independent. Case 3c percolates with probability 0.

This would result in the equation

$$\chi_t(r,s) = \frac{1}{r+s-t}\left[t + (r-t)\chi_t(r',s) + \frac{s-t}{1-r'-r}\left(t + (1-r')\chi_t(r,s')\right)\right].$$

However, this equation depends on r' and s', which are not defined, making the entire expression nonsense. However, in case 2, r' is chosen to be uniformly random on $[t,r]$, so the correct value of the probability of percolation is the average probability over this interval, namely $\frac{1}{r-t}\int_t^r \chi_t(r',s)dr'$. In case 3, r' is chosen to be uniformly random on the interval $[1+t-s, 1]$ and in case 3b, s' is chosen to be uniformly random on the interval $[t, 1+t-r']$. Thus the probability of percolation

in case 3b is $\frac{1}{1-r'}\int_t^{1+t-r'}\chi_t(r,s')ds'$ and the probability of percolation in case 3 is $\frac{1}{s-t}\int_{1+t-s}^{1}\left(t+\int_t^{1+t-r'}\chi_t(r,s')ds'\right)dr'$. So the total probability of percolation is

$$\begin{aligned}\chi_t(r,s) &= \frac{1}{r+s-t}\Big[t+\int_t^r \chi_t(r',s)dr' \\ &\quad +\frac{1}{1-r'-r}\int_{1+t-s}^{1}\left(t+\int_t^{1+t-r'}\chi_t(r,s')ds'\right)dr'\Big]\end{aligned}$$

which becomes

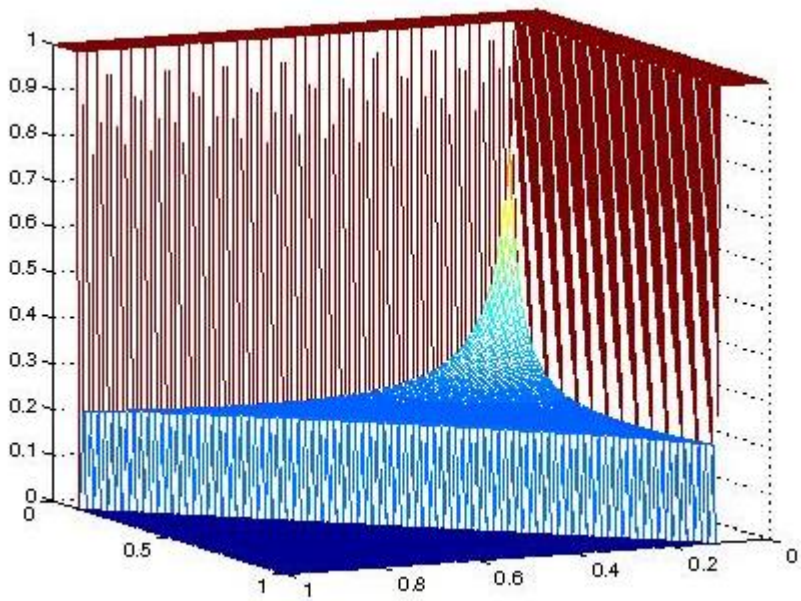
$$\chi_t(r,s)=\frac{1}{r+s-t}\left[t+\int_t^r \chi_t(r',s)dr'+\int_t^s \frac{1}{r+r'-t}\left(t+\int_t^{r'}\chi_t(r,s')ds'\right)dr'\right]$$

with a change of variables.

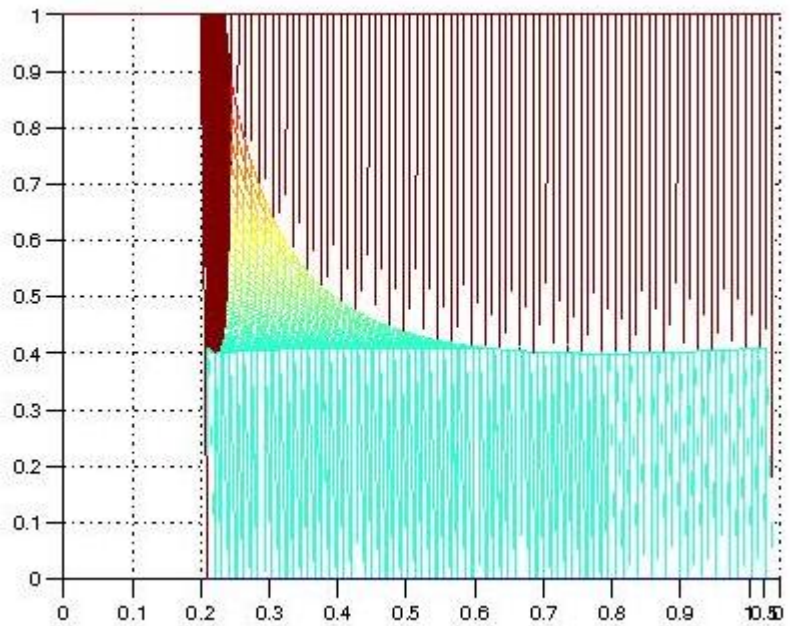
This analysis depends on successive iterations of the algorithm being independent. This may not be the case. Although values of b beyond i_2' are never examined, if case 3 is called the values a_i with i between i_1 and the last value of i_1' have already been examined, so it would seem that these values are dependent. This is true. Fortunately, none of these values matter. We have the following Lemma:

Lemma 3.5.1. *Any value in of a_i which has already been examined will not trigger one of the cases in this process.*

Proof. Consider an i in the range (i_1, i_1'). Since $i_1' > i$, the value a_i did not fall in the set R, defined previously. In other words, $a_i \in (r, 1+t-s)$. A value s' has been found in the range $[t, 1+t-r']$ since the algorithm is running again. But $s' < 1+t-r'$ implies that $s' < s$ so in the new iteration of the algorithm all values that were previously ignored (specifically, a_i) are ignored again. This range of ignored values is



(a) $\chi_t(r, s)$ with $t = .1$



(b) $\chi_t(r, s)$ with $t = .1$

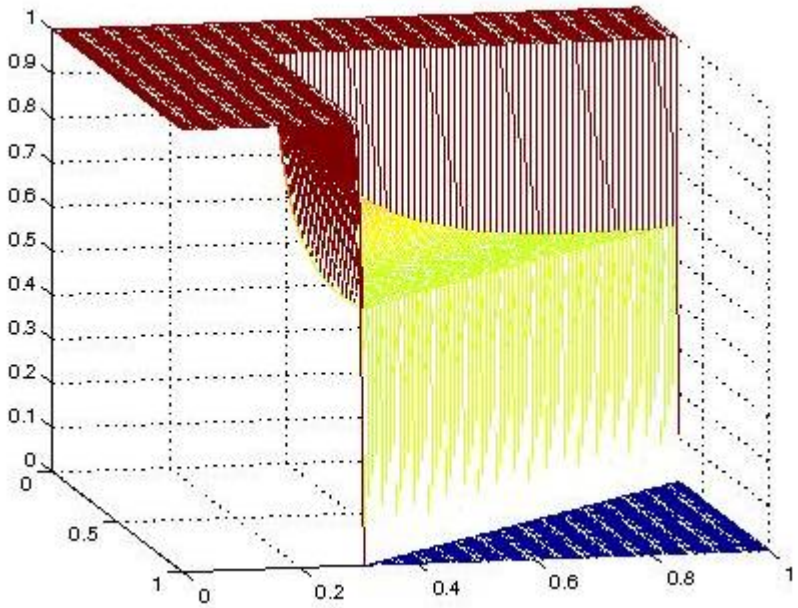
Figure 3.2: Examples of χ_t with (a) $t = .1$ and (b) $t = .2$.

now $(r, 1+t-s')$ and includes, in addition, r', so the new indicator i_1' will increment at least one farther than in the previous iteration of the algorithm. $\square$

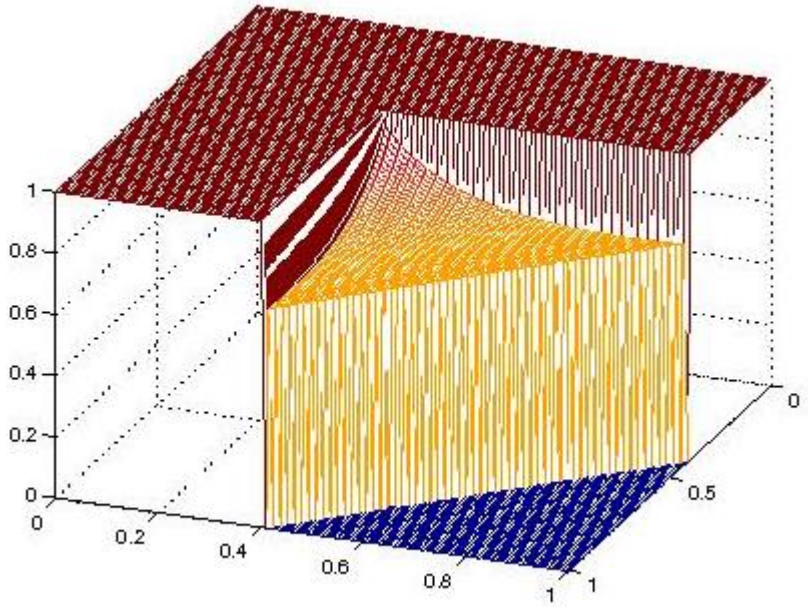
Therefore, any values that trigger one of the cases will be independent in all iterations. So the probability of percolation is given by the integral equation above.

Unfortunately, our attempts to find an explicit solution to this integral equation have so far failed. Figures 3.2 and 3.3 show graphs of $\chi_t(r,s)$ for $r, s \in [0,1]$ from various points of view. Notice that each of these displays the following characteristics: $\chi_t(r,s) = 1$ when r or s is less than t, $\chi_t(r,s) = 0$ when $r+s > 1+t$, the remaining values of r, s have a mostly constant value of $\chi_t(r,s)$, but it increases sharply as r or s approaches t. In Section 4.2, we will show $\theta(t)$ is analytic except at $t = 0$, and continuous at $t = 0$, from which it will follow that $\chi_t(r,s)$ is continuous except at the line $r+s = 1+t$. The continuity of the function is lost in these discrete pictures, however.

This process will be revisited in Section 6.1, where it is compared to other equivalent processes and leads to our main result in Section 6.3.



(a) $\chi_t(r, s)$ with $t = .3$



(b) $\chi_t(r, s)$ with $t = .4$

Figure 3.3: Examples of χ_t with (a) $t = .3$ and (b) $t = .4$.

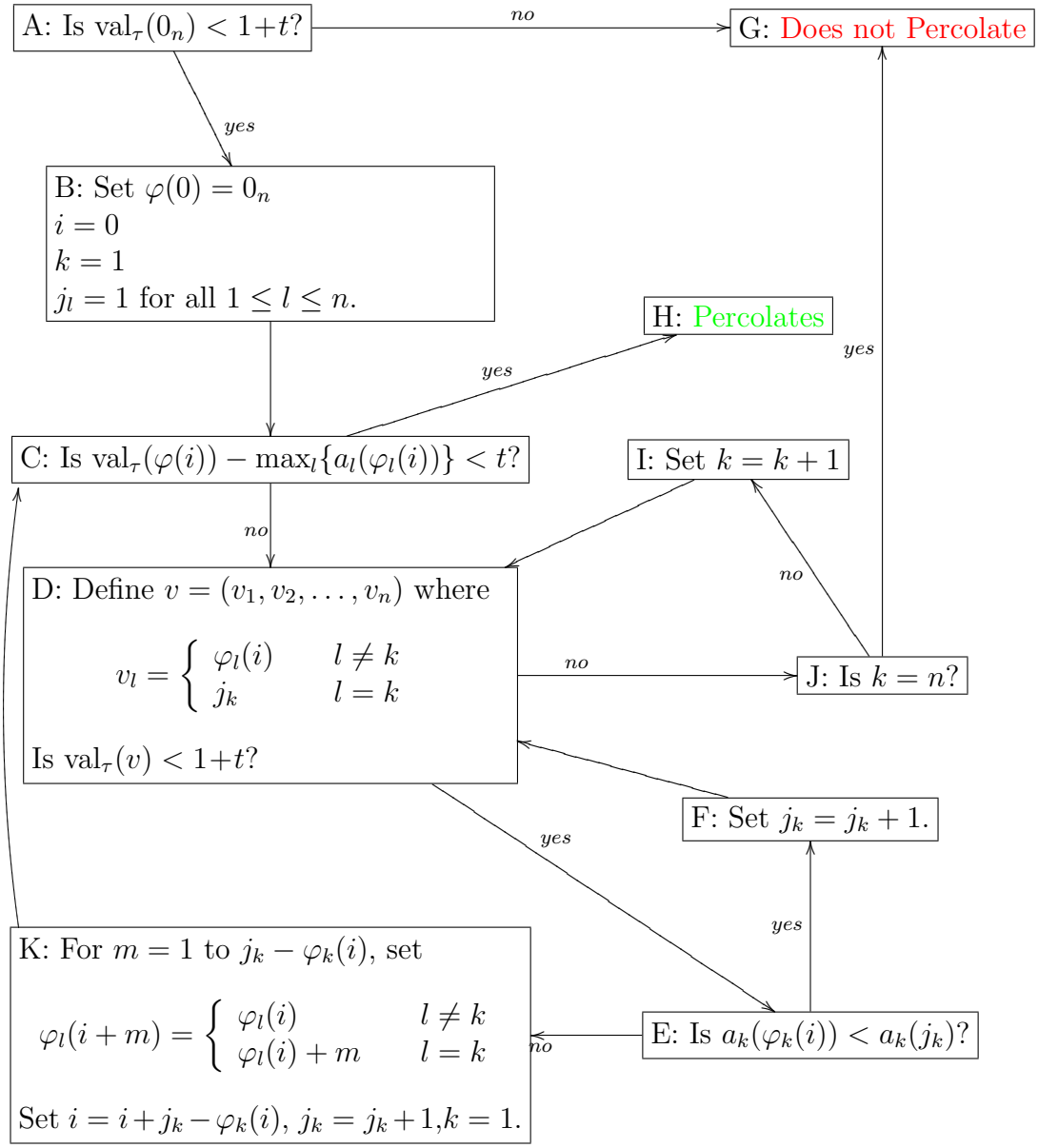


Table 3.1: A flowchart describing an algorithm without backtracking to determine whether the given instance $\tau \in \Omega_n$ percolates.

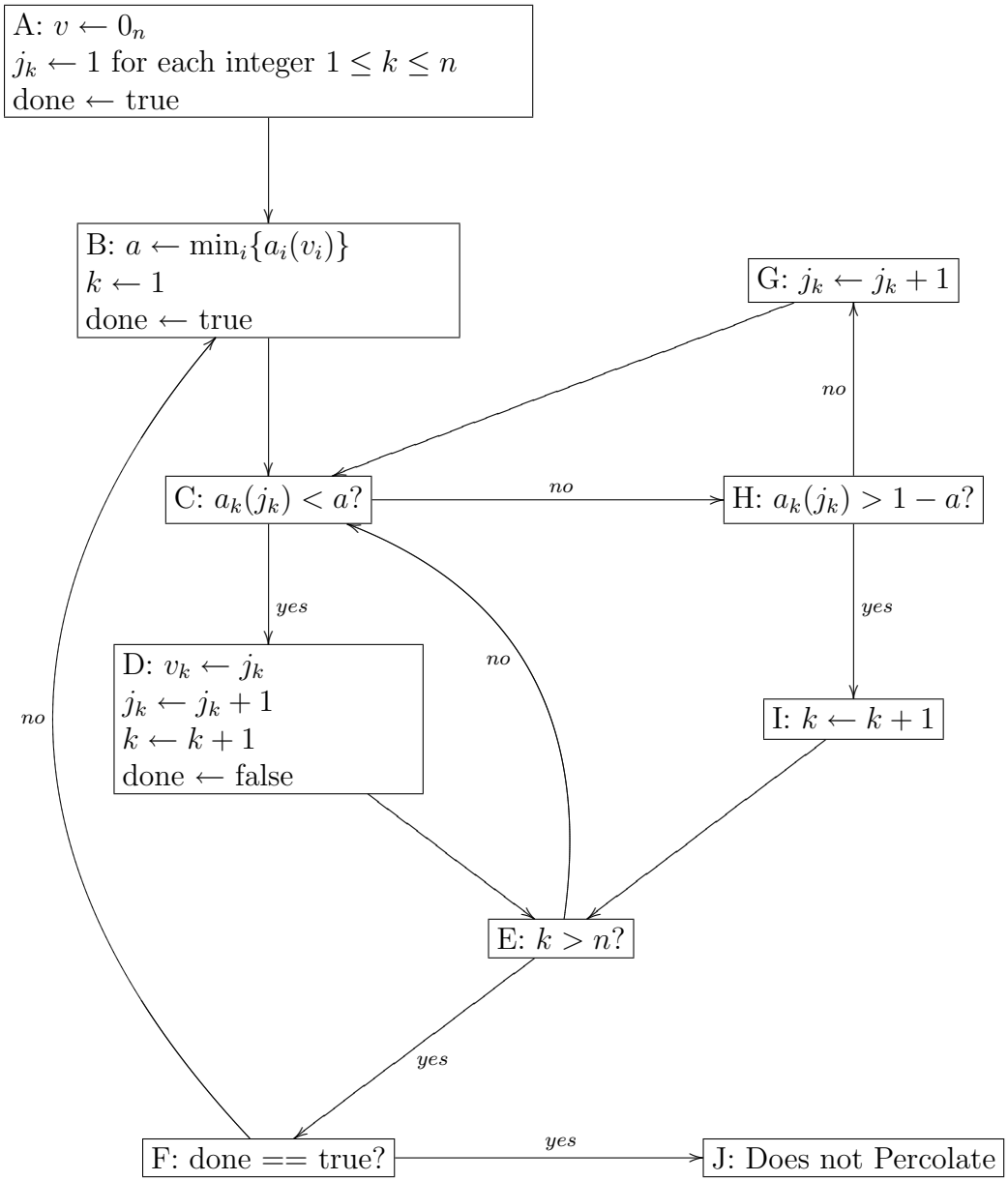


Table 3.2: A flowchart describing the algorithm to show that there are no $\tau \in \Omega_n$ which lead to a configuration $f_0(\tau) \in \Omega_n(0)$ which percolates.

Chapter 4

Combinatorial Approach

In the algorithmic approach, we generate and store a limited number of the usually infinite number of random variables at any one time, answering, for each allowed configuration in $\mathcal{F}_n$, the decision question, "Is percolation possible in this configuration?" The probability question, "What is the probability, $\theta(t)$, of percolation?" seems more difficult to tackle with combinatorics, because there are infinitely many random variables in each instance $\tau \in \Omega_n$, and each of these random variables has a continuous distribution. To start counting, it is first necessary to limit the problem, reducing the size of the objects of which we keep track, as we do in Section 4.1. This leads to a more tractable formula for the percolation probability, $\theta(t)$ in terms of a supplemental probability function $P(m, n)$. We are able to show this formula for $\theta(t)$ is analytic in Section 4.2, except at $t = 0$. In Section 4.3, we compute a recursive formula for $P(m, n)$.

4.1 Limiting the Problem

What do we mean by limiting the problem? Recall from Section 2.2 that $\theta(t)$ is defined as the probability that a uniformly random $\sigma \in \Omega_n$ has the property that $\Theta_t(\sigma) = 1$. There is an additional space $\mathcal{F}_n$ and a special function $\widetilde{\Theta_t} \colon \mathcal{F}_n \to \{0,1\}$ defined so that the following diagram commutes:

$$\begin{tikzcd}
\Omega_n \arrow[rr, "\Theta_t"] \arrow[dr, "f_t"'] & & \{0,1\} \\
& \mathcal{F}_n \arrow[ur, dotted, "\widetilde{\Theta_t}"'] &
\end{tikzcd}$$

Recall the function $\rho_t \colon \mathcal{F}_n \to [0,1]$ is defined by

$$\rho_t(\sigma) = \Pr[f_t(\tau) = \sigma]$$

for a uniformly random $\tau \in \Omega_n$. It is clear that

$$\theta(t) = \sum_{\sigma \ni \widetilde{\Theta_t}(\sigma)=1} \rho_t(\sigma).$$

However, the complexity of the space $\mathcal{F}_n$ makes computing $\rho_t(\sigma)$ difficult, and the size of the space makes computing the sum difficult. Therefore, we would like to have a new space A along with a function $f \colon \Omega_n \to A$ such that Θ_t is constant on $f^{-1}(\sigma)$ for every $\sigma \in A$. These conditions are sufficient so that there is a (unique) function $\tilde{f}$ so that the following diagram commutes:

$$\begin{tikzcd}
\Omega_n \arrow[rr, "\Theta_t"] \arrow[dr, "f"'] & & \{0,1\} \\
& A \arrow[ur, dotted, "\tilde{f}"'] &
\end{tikzcd}$$

Given such a pair A, f, define $p_A \colon A \to [0,1]$ by

$$p_A(\sigma) = \Pr[f(\tau) = \sigma]$$

for a uniformly random $\tau \in \Omega_n$. Then

$$\theta(t) = \sum_{\sigma \ni \tilde{f}(\sigma)=1} p_A(\sigma).$$

Although this analysis holds for general n, we only apply it in the case $n = 2$. Suppose $a_m < t$. It follows that all points (m, j) are open, for any j, and if any one of these points is accessible, then following the line $x = m$ is a path to infinity. In this case, the values of the random variables $a_{m+1}, a_{m+2}, \ldots$ do not affect the accessibility of the points (m, j), so they may be safely discarded. Similarly, if $b_n < t$, then the values $b_{n+1}, b_{n+2}, \ldots$ may be ignored.

In this way, if m is the first value so that $a_m < t$ and n is the first value so that $b_n < t$, there are only $m+n$ independent random variables to consider, each with uniform distribution on the interval $[t, 1]$. Define a space $\Omega_t(m, n)$ as the set of pairs of functions $(a \colon \{0, 1, \ldots, m-1\} \to [t, 1], b \colon \{0, 1, \ldots, n-1\} \to [t, 1])$. Let $A = \{(m, n, \sigma) |\ m, n \in \mathbb{N}, \sigma \in \Omega_t(m, n)\}$. The obvious map $f \colon \Omega_2 \to A$ is the map $(a, b) \mapsto (m, n, (a|_{\{0,1,\ldots,m-1\}}, b|_{\{0,1,\ldots,n-1\}}))$ where m, n are defined as above.

The values of m, n may also be considered as independent random variables, each having a geometric distribution with parameter t. In other words, for any $k \in \mathbb{N}$, $\Pr[m = k] = t(1-t)^k$. So $\sum_\sigma p_A(m, n, \sigma) = t^2(1-t)^{m+n}$ where the sum is over all of these reduced configurations. Unfortunately, we must still consider

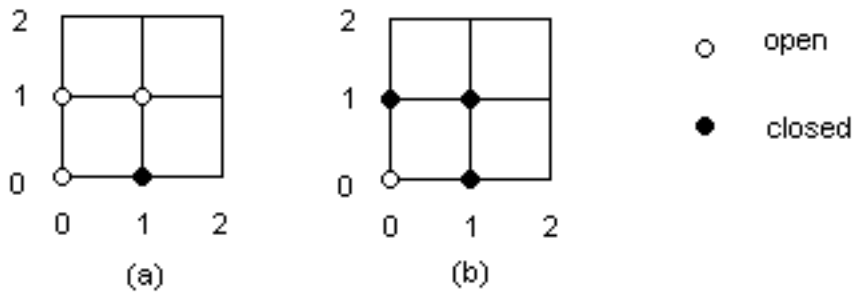


Figure 4.1: (a) The configuration $(2, 2, \{a_0 = .5, a_1 = .7, b_0 = .5, b_1 = .2\})$. (b) The configuration $(2, 2, \{a_0 = .5, a_1 = .7, b_0 = .5, b_1 = .7\})$. Notice that when $t = .1$, the points are open and closed as indicated, so (a) percolates and (b) does not. Recall the lines $x = 2, y = 2$ are open in both cases.

all the reduced configurations separately because it is easy to find examples where $\tilde{f}(m, n, \sigma) \neq \tilde{f}(m, n, \tau)$. See Figure 4.1 for an example in the case where $m = n = 2$.

Fix a configuration $\sigma \in \Omega_2$. Assume it satisfies the following properties:

1. There is a known path from a given point, (m, n) to infinity.

2. For every $i < m$, $a_i > t$ and the point (i, n) is open.

3. For every $j < n$, $b_j > t$ and the point (m, j) is open.

Let the probability of percolation in this setup be given by $P(m, n)$. It is easiest to think of $P(m, n)$ as the probability of percolation if the lines $x = m$ and $y = n$ are the first open lines. It is clear that the conditions of the definition are satisfied if these are the first open lines, although the definition is weaker than the condition that these lines are open. With this definition in hand, the probability of percolation

may be calculated as

$$\theta(t) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} P(m,n)t^2(1-t)^{m+n} \tag{4.1.1}$$

since the condition that $x = m$ and $y = n$ are the first open lines divides the sample space into mutually exclusive events. This reduces the problem to computing $P(m,n)$ for each pair (m,n) if $P(m,n)$ is independent of t, which will be shown in Theorem 4.1.1. This proof relies on the fact that for any $\sigma \in S_n$ and independent, identically distributed continuous random variables $X_1, X_2, \ldots, X_n$, the probability that $X_{\sigma(1)} < X_{\sigma(2)} < \cdots < X_{\sigma(n)}$ is $1/n!$. Define an involution $\rho\colon \Omega_t(m,n) \to \Omega_t(m,n)$ by $(a,b) \mapsto (a,c)$ where $c\colon \{0,1,\ldots,n-1\} \to [t,1]$ is defined by

$$c(i) = 1 + t - b(i).$$

Abbreviate $c(i)$ as c_i. The condition of an open point, $a_i + b_j < 1 + t$ is equivalent to the condition $a_i < c_j$. We have lost the information about whether points outside of the box $[m] \times [n] \subset \mathbb{N}^2$ are open and closed, but this does not affect percolation. Define a new configuration space $T_t(m,n) = \{\sigma\colon [m] \times [n] \to \{0,1\}\}$ and a map $h_t\colon \Omega_t(m,n) \to T_t(m,n)$ by

$$h_t(\sigma)(i,j) = 1 \Leftrightarrow a_i < c_j$$

where the a_i, c_j are taken from the instance σ.

This redefinition of the problem allows us to look at the words $a = a_0a_1a_2\ldots a_{m-1}0$

Integer Index in σ	Corresponding Line
1	$x = 0$
2	$x = 1$
$\vdots$	$\vdots$
$i + 1$	$x = i$
$\vdots$	$\vdots$
m	$x = m - 1$
$m + 1$	$y = 0$
$\vdots$	$\vdots$
$m + i + 1$	$y = i$
$\vdots$	$\vdots$
$m + n$	$y = n - 1$

Table 4.1: The correspondence between the lines in $\sigma \in \Omega_t(m, n)$ and the integer inputs to the permutation $\tau \in S_{m+n}$ where $g_t(\sigma) = \tau$.

and $c = c_0c_1c_2 \dots c_{n-1}1$. The condition of an open path from $(0, 0)$ to (m, n) is equivalent to the condition $a \preceq c$ in worm order. For information on worms and worm order, see Chapter 5 or [2].

Theorem 4.1.1. *Define $P_t(m, n)$ as the probability that a uniformly random configuration $\sigma \in \Omega_2$ satisfying the conditions $a_i > t, b_j > t$ for all $0 \leq i < m$, $0 \leq j < n$ and $a_m < t, b_n < t$ has the property that $\Theta_t(\sigma) = 1$. Then $P_t(m, n) = P_s(m, n)$ for all $0 < s, t < 1$.*

Proof. For each t, define a map $g_t \colon \Omega_t(m, n) \to S_{m+n}$ (where S_n is the permutation group on n elements) by $(a, c) \mapsto \sigma$ if the set of values of a and c is $\{x_1 < x_2 < \cdots < x_{m+n}\}$ and $\sigma(i) = j$ when $i \leq m$ and $a_{i-1} = x_j$, or $i > m$ and $c_{i-m-1} = x_j$. (See Table 4.1 to see the correspondence between the integer index in the permutation σ and the corresponding line in the grid $[m] \times [n]$.) Since the values of a_i and c_j are

independent, identically distributed continuous random variables, $\Pr[g_t(\sigma) = \tau]$ for a uniformly random $\sigma \in \Omega_t(m,n)$ and a specified $\tau \in S_{m+n}$ is exactly $1/(m+n)!$. Define a map $p_t \colon S_{m+n} \to T_t(m,n)$ as follows: for $\sigma \in S_{m+n}$, $p_t(\sigma)(i,j) = 1$ if and only if $\sigma(i+1) < \sigma(j+m+1)$. The map p_t also depends on m, n but these values should be clear from context. Notice for $\sigma \in \Omega_t(m,n)$, $p_t(g_t(\sigma))(i,j) = 1$ if and only if $g_t(\sigma)(i+1) < g_t(\sigma)(j+m+1)$ if and only if $a_i < c_j$ since $a_i = x_{g_t(\sigma)(i+1)}$ and $c_j = x_{g_t(\sigma)(j+m+1)}$ by the construction of $g(t)$ and $x_k < x_l$ if and only if $k < l$. Therefore, $p_t \circ g_t = h_t$.

Let $C = \{(m,n,\sigma) |\ m,n \in \mathbb{N}, \sigma \in T_t(m,n)\}$ and consider the following commutative diagram:

$$\begin{tikzcd}
\Omega_2 \arrow[rr, "\Theta_t"] \arrow[dd, "f"'] \arrow[drrr, dotted, "\xi"] & & & \{0,1\} \\
& & C \arrow[ur] & \\
A \arrow[rr, "(m,n,(a,b))\mapsto(m,n,\rho(a,b))"] & & A \arrow[u, "(m,n,\sigma)\mapsto(m,n,h_t(\sigma))"] \arrow[r, "(m,n,\sigma)\mapsto(m,n,g_t(\sigma))"] & B \arrow[uu, dotted, "\hat{f}"'] \arrow[ul, "(m,n,\sigma)\mapsto(m,n,p_t(\sigma))"']
\end{tikzcd}$$

where $\hat{f}$ is defined to make the diagram commute and $\xi \colon \Omega_2 \to B$ is the composition map. However, $P_t(m,n) = \Pr[\Theta(\sigma) = 1 | f(\sigma) = (m,n,\sigma')]$ for some $\sigma' \in \Omega_t(m,n)$. Thus

$$P_t(m,n) = \sum_{\sigma \in S_{m+n}} \Pr[\hat{f}(m,n,\sigma) = 1].$$

The right hand side of this equation depends only on m, n and the group S_{m+n}: it

is independent of t. Therefore, for every s, t, $P_t(m,n) = P_s(m,n)$. $\square$

4.2 Analyticity

With a more combinatorial interpretation of the problem, and this formula for computing $\theta(t)$, some facts are more readily obtainable than with the algorithmic approach. The following theorem is one of the more immediate consequences.

Theorem 4.2.1. *$\theta(t)$ is analytic for $0 < t \leq 1$.*

Proof. The formula in (4.1.1) can be rewritten as a power series in the variable $z = 1 - t$ as

$$\theta(t) = \sum_{m=0}^{\infty} c_m z^m \tag{4.2.1}$$

where

$$\begin{aligned} c_0 &= 1, \\ c_1 &= 0, \text{ and} \\ c_m &= \sum_{i=0}^{m-2} P(i, m-2-i) - 2\sum_{i=0}^{m-1} P(i, m+1-i) + \sum_{i=0}^{m} P(i, m-1) \text{ for } m > 1. \end{aligned} \tag{4.2.2}$$

Since $\theta(0) = \sum\sum 0^2(1)^{m+n}P(m,n) = 0$, this is clearly convergent to 0 at $z = 1$, and so has a radius of convergence of at least 1. Since c_m is a finite sum, it is convergent, and derivatives of convergent series have the same radius of convergence as the original function. Therefore, this function is analytic. (For proofs of these

statements, see any complex analysis textbook. For example, [3].) This gives the desired range of t. $\square$

This theorem emphasizes the approachability of this model of percolation, as the question of whether or not the probability function is analytic is still not answered in many other models. The theorem is not complete, however, since it does not address differentiability at $t = 0$.

To address differentiability of $\theta(t)$ at $t = 0$, consider the following very limited algorithm for percolation. For a given $\sigma \in \Omega_2$, find the smallest m so that $x_m < t$. If all of the points $(i, 0)$ with $i \leq m$ are open, we percolate. Given m, the probability of percolation by this algorithm is exactly $1/(m+1)$ since the variables considered are $c_0, a_0, a_1, \ldots, a_{m-1}$ and this algorithm will reveal percolation if and only if c_0 is the largest of these $m+1$ independent, identically distributed continuous random variables. Thus the total probability of percolation by this algorithm is

$$\sum_{i=0}^{\infty} \frac{t(1-t)^m}{m+1}.$$

$\theta(t)$ must be larger than this quantity. Therefore, for every t we have the ratio

$$\frac{\theta(t)}{t} > \frac{\sum_{i=0}^{\infty} \frac{t(1-t)^m}{m+1}}{t} = \sum_{i=0}^{\infty} \frac{(1-t)^m}{m+1}$$

so the derivative of $\theta(t)$ at $t = 0$, defined as

$$\begin{aligned}
\lim_{t\to 0} \frac{\theta(t) - \theta(0)}{t - 0} &= \lim_{t\to 0} \sum_{m=0}^{\infty} \frac{(1-t)^m}{m+1} \\
&= \lim_{t\to 0} - \int \sum_{m=0}^{\infty} (1-t)^m dt \\
&= \lim_{t\to 0} - \int \frac{1}{1-(1-t)} dt \\
&= \lim_{t\to 0} - \ln(t)
\end{aligned}$$

does not exist. Therefore, $\theta(t)$ is not differentiable at $t = 0$. In Chapter 6, we prove a formula for $\theta(t)$ in the case $n = 2$.

Recall in Section 3.5 we defined $\chi_t(r, s)$ as the probability of percolation given that $a_0 = r$ and $b_0 = s$. We now prove this function is continuous. The idea of the proof is to break down $\chi_t(r, s)$ as the sum of $P(m, n, r, s)$, similar to how we have broken down $\theta(t)$ into the sum of $P(m, n)$, and to show that the $P(m, n, r, s)$ are continuous. This is due to the fact that it is very unlikely that the change of r to $r + \delta$ is very unlikely to result in a change of the configuration.

Theorem 4.2.1. *$\chi_t(r, s)$ is continuous except at the line $r + s = 1 + t$.*

Proof. Let $P(m, n, r, s)$ be the probability of percolation for an instance $(a, b) \in \Omega_2$ with $a_0 = r, b_0 = s$, and the first open lines in each direction being $x = m$ and $y = n$. Then

$$\chi_t(r, s) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} P(m, n, r, s) t^2 (1-t)^{m+n}.$$

Thus, if $P(m, n, r, s)$ is continuous for every m, n, r, s then $\chi_t(r, s)$ is continuous for every r, s. However, $P(m, n, r, s)$ and $\chi_t(r, s)$ are not continuous at the line

$r + s = 1 + t$, so we omit this case. For r, s with $r + s > 1 + t$, $\chi_t(r, s) = 0$, so it is continuous for these values. Assume $r + s < 1 + t$.

Let $\varepsilon > 0$. Pick δ so that $\varepsilon > 1 - (1 - \delta)^{m+n-1}$. Select an instance $\sigma = (a, b) \in \Omega_t(m, n)$ with $a_0 = r$ and $b_0 = s$. Let σ' be the instance which is σ with a_0 replaced with $r+\delta$ assuming $r+\delta+s < 1+t$. The difference $|P(m, n, r+\delta, s)-P(m, n, r, s)| < \Pr[g_t(\sigma) \neq g_t(\sigma')$. But $g_t(\sigma) = g_t(\sigma')$ unless there is some i or j so that $a_i \in (r, r+\delta)$ or $b_j \in (1 + t - (r + \delta), 1 + t - r)$. The probability that any one of these $m + n - 1$ values is outside this range is $1-\delta$ so the probability that all of them are outside the range is $(1-\delta)^{m+n-1}$. Therefore, the probability that g_t is different is $1-(1-\delta)^{m+n-1}$ and the difference $|P(m, n, r + \delta, s) - P(m, n, r, s)| < 1 - (1 - \delta)^{m+n-1} < \varepsilon$. Thus $P(m, n, r, s)$ is continuous in r. A similar argument shows $P(m, n, r, s)$ is continuous in s, and this results in $\chi_t(r, s)$ being continuous in r and s, as desired. $\square$

4.3 Computing $P(m, n)$

Since the line $y = 0$ being open already creates an infinite open cluster containing the origin, we know $P(m, 0) = 1$. A similar argument shows $P(0, n) = 1$. This seems to be a good place to start a recursion, and we find the following:

Theorem 4.3.1. *Given m, n,*

$$P(m, n) = \frac{1}{m + n}\left[\sum_{i=2}^{m}\sum_{j=1}^{i-1}\frac{1}{n + i - 1}P(j - 1, n) + \sum_{i=m+1}^{m+n} P(m, i - m - 1).\right]$$

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	1	0.5	0.45833	0.44167	0.43333	0.42857	0.42560	0.42361	0.42222	0.42121
3	1	0.5	0.44167	0.41528	0.40099	0.39236	0.38674	0.38287	0.38009	0.37803
4	1	0.5	0.43333	0.40099	0.38256	0.37099	0.36322	0.35774	0.35372	0.35069
5	1	0.5	0.42857	0.39236	0.37099	0.35718	0.34770	0.34088	0.33580	0.33191
6	1	0.5	0.42560	0.38674	0.36322	0.34770	0.33684	0.32891	0.32293	0.31830
7	1	0.5	0.42361	0.38287	0.35774	0.34088	0.32891	0.32006	0.31332	0.30805
8	1	0.5	0.42222	0.38009	0.35372	0.33580	0.32293	0.31332	0.30593	0.30010
9	1	0.5	0.42121	0.37803	0.35069	0.33191	0.31830	0.30805	0.30010	0.29379
10	1	0.5	0.42045	0.37646	0.34835	0.32887	0.31464	0.30385	0.29542	0.28869

Table 4.2: Values of $P(m, n)$

Proof. Let $m, n \in \mathbb{N}$. Recall from Theorem 4.1.1 that

$$P(m, n) = \sum_{\sigma \in S_{m+n}} \Pr[\hat{f}(m, n, \sigma) = 1].$$

Define a map $p_t \colon S_{m+n} \to \{0, 1\}$ by

$$p_t(\sigma) = 0 \Leftrightarrow \hat{f}(m, n, \sigma) = 0.$$

Therefore,

$$P(m, n) = \sum_{\sigma \in S_{m+n}} \Pr[p_t(\sigma) = 1].$$

Let $\sigma \in S_{m+n}$. Condition on the value $i = \sigma^{-1}(m + n)$.

If $i = 1$, the correspondence in Table 4.1 says that this corresponds to the line $x = 0$. Therefore, in $p_t(\sigma)$, the points $(0, i)$ are closed ($p_t(\sigma)(0, i) = 0$ since $\sigma(1) >$

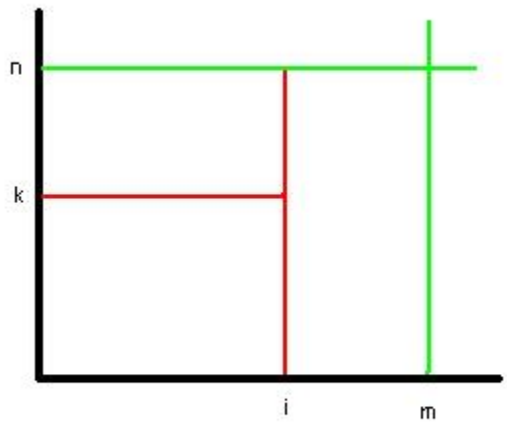


Figure 4.2: Since $\sigma(i) = m + n$ and $\sigma(j) < \sigma(k)$ for every $k < i$, we have the points on the north and east boundaries of the interior box are closed.

$\sigma(m+i+1))$ for $0 \leq i < n$. In particular, the point $(0, 0)$ is closed so this configuration does not percolate.

If $2 \leq i \leq m$, we have $\sigma(i) > \sigma(j)$ for every $m + 1 \leq j \leq m + n$. Thus $p_t(\sigma)(i - 1, j - m - 1) = 0$. Define

$$I(\sigma, i) = \begin{cases} \{\sigma(j) |\ j < i \text{ or } j > m\} & i \leq m \\ \{\sigma(j) |\ j < i\} & i > m \end{cases}$$

Let $j = \sigma^{-1}(\min I(\sigma, i))$ be the integer mapping to the minimum element of $I(\sigma, i)$. If $j > m$, we are in the case illustrated in Figure 4.2. Since $\sigma(j) < \sigma(k)$ for every $k < i$, all of the points $(k - 1, j - m - 1)$ are closed. These and the closed points

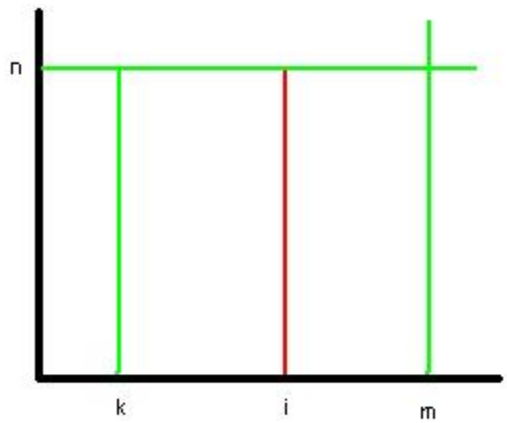


Figure 4.3: Since $\sigma(i) = m + n$, we have $\sigma(j) < \sigma(k)$, for every $k > m$, the points on the line $x = k$ with $y \leq n$ are open.

$(i-1, k-m-1)$ with $k > m$ form a closed box around the origin. Thus the configuration $p_t(\sigma)$ does not percolate. Otherwise, $j < i$ and we are in the case illustrated in Figure 4.3. Since $\sigma(j) < \sigma(k)$ for every $k > m$ all of the points $(j-1, k-m-1)$ are open. Reduce $\sigma \in S_{m+n}$ to a permutation $\tau \in S_{j+n}$ as follows: Define a map $l\colon \{1, 2, 3, \ldots, j+n\} \to \{1, 2, \ldots, j, m+1, m+2, \ldots, m+n\}$ by

$$l(k) = \begin{cases} k & k \leq j \\ k + m - j & k > j \end{cases}$$

Notice the map o is bijective. Define a map $q\colon I(\sigma, j) \to \{1, 2, \ldots, j+n\}$ by $q(x_k) = k$ where $I(\sigma, j) = \{x_1 < x_2 < \cdots < x_{j+n}\}$. Notice this map is also bijective. Let

$\tau = q \circ \sigma \circ l$. Since σ is a bijection between $\{1, 2, \ldots, j, m+1, m+2, \ldots, m+n\}$ and $I(\sigma, j)$, this entire map is a bijection of the sets it is defined on, namely $\{1, 2, 3, \ldots, j+n\}$. Therefore, $\tau \in S_{j+n}$. Since $\tau(k) < \tau(l)$ if and only if $\sigma(o(k)) < \sigma(o(l))$, the configuration $p_t(\tau)$ is the same as the restriction of the configuration $p_t(\sigma)$ to the sublattice $[j] \times [n]$. Thus the probability of percolation in this case is exactly $P(j, n)$. The probability that j maps to the lowest element in $I(\sigma, j)$ is $1/(i+n)$.

Otherwise, $i > m$. This is the case illustrated in Figure 4.4. Here, $\sigma(i) > \sigma(k)$ for every $k > m$ and all of the points $(k-1, i-m-1)$ are open in $p_t(\sigma)$. We may restrict the permutation σ to a permutation of the integers $\{1, 2, \ldots, i-1\}$ by defining $q\colon I(\sigma, i) \to \{1, 2, \ldots, i-1\}$ by $q(x_k) = k$ where $I(\sigma, j) = \{x_1 < x_2 < \cdots < x_{m+i}\}$. Let $\tau = q \circ \sigma$. Since $\tau(j) < \tau(k)$ if and only if $\sigma(j) < \sigma(k)$, the configuration $p_t(\tau)$ is the same as the restriction of the configuration $p_t(\sigma)$ to the sublattice $[m] \times [i]$. Therefore the probability of percolation in this case is exactly $P(m, i)$.

To get the above sum, note that the choice of i occurs with probability $1/(m+n)$. When $1 < i \leq m$, we also need to choose a j with probability $1/(i+n)$ and then percolate with probability $P(j, n)$. When $i > m$, we percolate with probability $P(m, i)$. The choice of i and j (if applicable) divides the sample space into mutually exclusive events, so the total probability of percolation is the sum of these probabilities. In other words,

$$P(m, n) = \frac{1}{m+n} \left[\sum_{i=2}^{m} \sum_{j=1}^{i-1} \frac{1}{n+i-1} P(j-1, n) + \sum_{i=m+1}^{m+n} P(m, i-m-1). \right]$$

as desired.

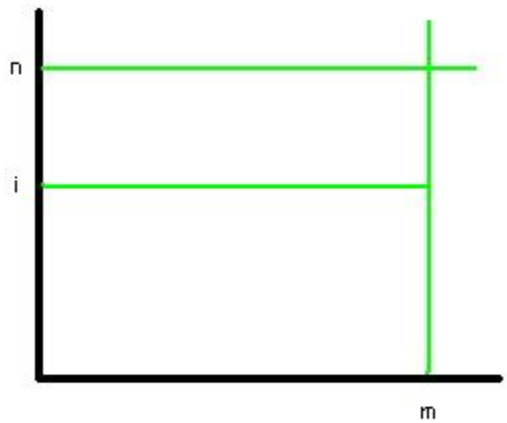


Figure 4.4: Since $\sigma(i) = m + n$, we have $\sigma(k) < \sigma(i)$, for every $k > m$, the points on the line $y = i$ with $x \leq m$ are open.

$\square$

4.4 A Closer Look

The correspondence to permutations in S_{m+n} to instances in Ω_2 is very predictable. Given m, n, the probability of a given permutation is $1/(m+n)!$ and the probability of a given pair m, n is exactly $t^2(1-t)^{m+n}$. Therefore, the probability that $\xi(\sigma) = (m, n, \tau)$ for a uniformly random $\sigma \in \Omega_2$ is exactly $t^2(1-t)^{m+n}/(m+n)!$. Why is it difficult to decide if this permutation percolates or not?

One answer is that while ξ is a predictable map, p_t is not. See Figure 4.5 to demonstrate this. In order to compute the probability of percolation, we must first

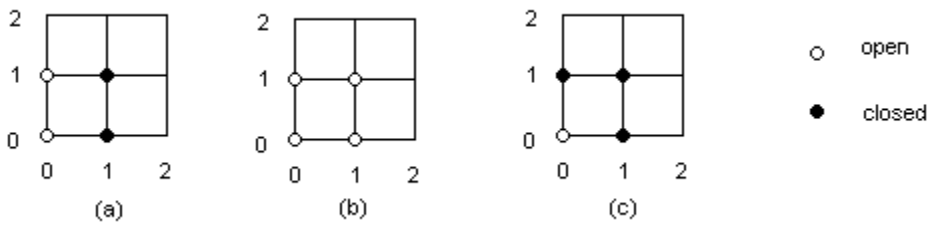


Figure 4.5: (a) This configuration corresponds to exactly 2 permutations in S_4: $(1,3,4,2),(1,4,3,2)$.(b) This configuration corresponds to exactly 4 permutations in S_4: $(1,2,3,4),(2,1,3,4),(1,2,4,3),(2,1,4,3)$. (c) This configuration corresponds to exactly 1 permutation in S_4: $(1,3,2,4)$.

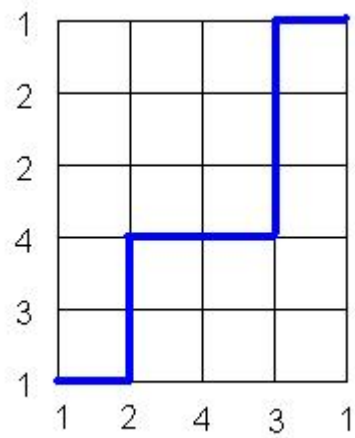
understand this correspondence, and then figure out which configurations percolate, or we must understand directly which configurations percolate.

Chapter 5

Worms

A worm, a combinatorial object introduced by Brightwell and Winkler [2], is relatively new and it is explored here. The proofs in this chapter use different techniques than the proofs in the paper. The set-up and definitions needed to solve the problems are given in Section 5.1. In Section 5.2, we answer the question "What is a worm?" and prove that this is well-defined. Section 5.3 studies the binary operation of addition on worms. This is used in the original application for worms, which is toward a scheduling problem. Section 5.4 studies two other operations, the meet and the join, and proves that, with these definitions, the partially ordered set of worms forms a lattice. Section 5.5 studies the number of worms on a finite (linearly ordered) alphabet. Section 5.6 gives the application to percolation.

Figure 5.1: A schedule of a versus b for $a = 1, 2, 4, 3, 1$ and $b = 1, 3, 4, 2, 2, 1$.



5.1 Definitions

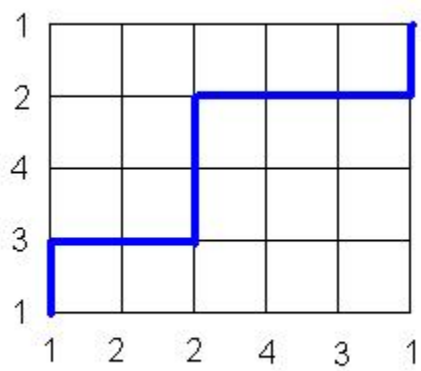
Suppose $a = a_0a_1 \dots a_m$, $b = b_0b_1 \dots b_n$ and $c = c_0c_1 \dots c_l$ are words on the alphabet of real numbers. Let ab be the word $a_0a_1 \dots a_mb_0b_1 \dots b_n$ and define $\mathrm{rev}(a)$ as the reverse of the word a, that is $\mathrm{rev}(a) = a_m \dots a_1a_0$. A **schedule of** a **versus** b is a path in the lattice $[m+1] \times [n+1]$ beginning at the point $(0, 0)$ and ending at (m, n) which advances only with steps to the north and the east. For an example of a schedule, see Figure 5.1. As is done in this figure, the points of the lattice will usually be labelled with the letters in the words rather than the integers.

For any schedule φ of a versus b, there is an associated **transpose schedule** φ^T of b versus a obtained by flipping the image across the diagonal $x = y$.

Figure 5.2 shows the transpose schedule associated to the schedule in Figure 5.1.

We say that a **precedes** b **in worm order**, or $a \preceq b$, if there is a schedule of a versus b so that $a_i \leq b_j$ for all points (i, j) on the path. A scheduling for which

Figure 5.2: The transpose schedule associated to the schedule of a versus b in Figure 5.1.



this holds is called a **witness**. Figure 5.3 shows an example witness of $a \preceq b$ for the given two words a and b. Although we will show shortly that the relation $\preceq$ is a pre-order on real words, it is not anti-symmetric. Figure 5.4 gives an example witness of $b \preceq a$ for the same two words a and b as in Figure 5.3. This is just one of the many examples showing that $\preceq$ is not anti-symmetric.

Before showing that $\preceq$ is a pre-order, we can list some simple properties that it tells us about the words it compares.

Theorem 5.1.1. *If $a \preceq b$, then the following are true:*

1. $a_0 \leq b_0$
2. $a_m \leq b_n$
3. $\max\{a_i\} \leq \min\{b_j\}$
4. $\min\{a_i\} \leq \min\{b_j\}$

Figure 5.3: A witness of $a \preceq b$ for a and b as in Figure 5.1.

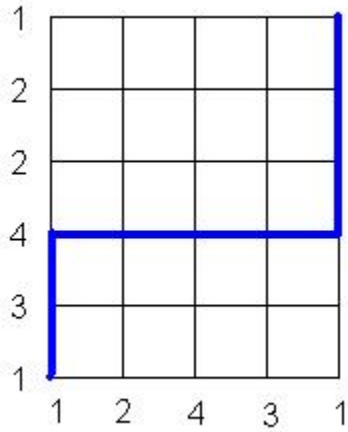
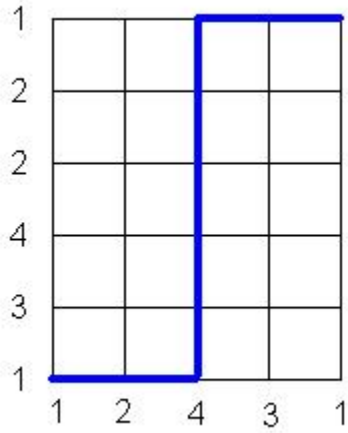


Figure 5.4: A witness of $b \preceq a$ for a and b as in Figure 5.1.



Proof. Let φ be a witness of $a \preceq b$. For any scheduling, we begin at $(0,0)$ and end at (m,n), so (1) and (2) are trivial. For (3), let $\hat{\imath}$ be the index of $\max\{a_i\}$. Since the lattice path must pass through the line $x = \hat{\imath}$, there is a $\hat{\jmath}$ so that $(\hat{\imath}, \hat{\jmath})$ is on φ. $\max\{a_i\} = a_{\hat{\imath}} \leq b_{\hat{\jmath}} \leq \max\{b_j\}$, as desired. (4) is shown similarly. $\square$

We say that $a \equiv b$ if $a \preceq b$ and $b \preceq a$.

Theorem 5.1.2. *The relation $\preceq$ is reflexive and transitive, so $\equiv$ is an equivalence relation on $\mathbb{R}^*$ and $\preceq$ forms a partial order on the equivalence classes of $\equiv$.*

Proof. **Reflexive:** Let a be a word. Define a lattice path by beginning at $(0,0)$. If at (i,i), move to $(i+1,i)$ if $a_{i+1} \leq a_i$ and move to $(i,i+1)$ otherwise. If not at (i,i) for some i, move either north or east until reaching this (it will only be one step away). This path is a witness of $a \preceq a$.

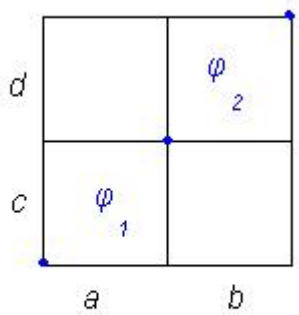
Transitive: Let φ_1 witness that $a \preceq b$ and φ_2 witness that $b \preceq c$. Create a witness θ of $a \preceq c$ by first creating a path Φ in $[m] \times [n] \times [l]$ so that for any point (i,j,k) on Φ it is true that (i,j) is on φ_1 and (j,k) is on φ_2. To create Φ, begin at $(0,0,0)$. When at (i,j,k), the next point is $(i+1,j,k)$ if $(i+1,j)$ is on φ_1. Otherwise, the next point is $(i,j,k+1)$ if $(j,k+1)$ is on φ_2. Otherwise, the next point is $(i,j+1,k)$.

Notice that the next point is still on both paths φ_1 and φ_2. The first case only moves the i coordinate, but φ_1 is moving in the i coordinate. The second case only moves the k coordinate, but φ_2 is moving in the k coordinate. The third case only moves the j coordinate, and both φ_1 and φ_2 are moving in the j coordinate.

From Φ, define θ as follows. Beginning at $(0,0)$, the steps taken on Φ define the steps taken on the new path. On Φ, suppose we're at (i,j,k). We will be at (i,k) on θ. If Φ goes to $(i+1,j,k)$, θ moves to $(i+1,k)$. If Φ goes to $(i,j,k+1)$, θ moves to $(i,k+1)$. Otherwise, Φ goes to $(i,j+1,k)$ and θ doesn't move.

For any (i,k) on θ, there is a j (not necessarily unique) so that (i,j,k) is on Φ, so (i,j) is on φ_1 and (j,k) is on φ_2. Thus $a_i \leq b_j \leq c_k$ and θ is a witness of $a \preceq c$,

Figure 5.5: A witness that $ac \preceq bd$.



as desired.

□

Lemma 5.1.1. *If a, b, c, d are words, $a \preceq b$, and $c \preceq d$ then $ac \preceq bd$.*

Proof. Suppose φ_1 is a witness of $a \preceq b$ and φ_2 a witness of $c \preceq d$. Construct a witness θ of $ac \preceq bd$ by concatenating the paths φ_1 and φ_2, as in Figure 5.5.

□

Lemma 5.1.2. *For words a, b with $a \preceq b$,* $\mathrm{rev}(a) \preceq \mathrm{rev}(b)$.

Proof. For any witness φ of $a \preceq b$, we can reverse the path to form a witness of $\mathrm{rev}(a) \preceq \mathrm{rev}(b)$. □

A word a is called **expanding** if each entry is a local maximum and bigger than all previous entries or a local minimum and smaller than all previous entries. In other words, for an entry a_i, one of the following is true:

- $a_i > a_{i+1}$ (if it exists) and $a_i > a_j$ for any $j < i$, or
- $a_i < a_{i+1}$ (if it exists) and $a_i < a_j$ for any $j < i$.

A word b is called **contracting** if $\mathrm{rev}(b)$ is expanding.

5.2 The Existence of Worms

The unique shortest representative of any equivalence class of $\equiv$ is called a **worm**. To show the uniqueness of the shortest representative, we first show that any shortest representative has a distinct form, given in Theorem 5.2.1, and then show that any two words of that form must be in different equivalence classes. Together, these say that there is only one shortest representative of any equivalence class, so worms are well-defined.

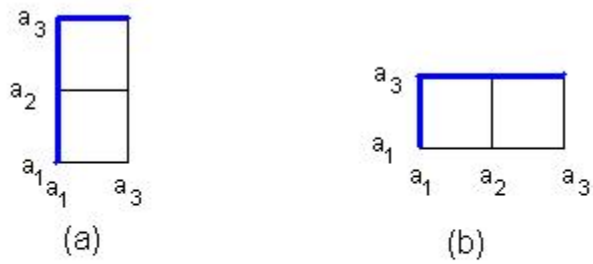
Theorem 5.2.1. *Suppose w is the shortest word in its equivalence class. Then $w = AcB$ where A and $\mathrm{rev}(B)$ are expanding worms and c is the final occurrence of the maximum letter in w.*

Proof. Suppose $w = AcB$ as in the statement of the theorem and w is the shortest word in its equivalence class. It is sufficient to show that A is an expanding word, since symmetry will give that $\mathrm{rev}(B)$ is expanding.

Suppose otherwise.

Case 1: A has a sequence of 3 increasing entries. In other words, $A = A_1 a_1 a_2 a_3 A_2$ where $a_1 \leq a_2 \leq a_3$. Let $A' = A_1 a_1 a_3 A_2$. Then $A \equiv A'$ (see Figure 5.6

Figure 5.6: Witnesses of (a)$A' \preceq A$ and (b)$A \preceq A'$ for A' as in Case 1.



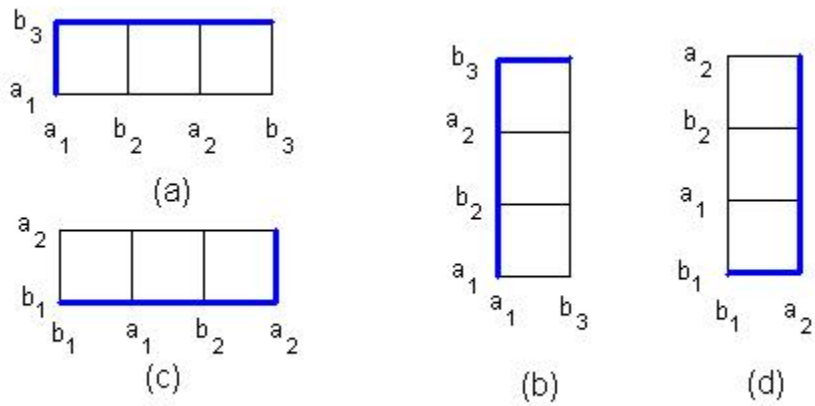
for witnesses) and w is not the shortest word in its equivalence class ($A'cB$ is shorter).

Case 2: A has a sequence of 3 decreasing entries. This is handled similarly to Case 1.

Case 3: A has an entry which repeats immediately. Since $aa \equiv a$, w cannot be the shortest word in its equivalence class.

Case 4: There is a local maximum which is not larger than all previous entries. Locate one local maximum so that it is smaller than both adjacent local maximums, and label this b_2 (Note: the adjacent local maximum on the right may be c, the global max.) Let $A = A_1 b_1 a_1 b_2 a_2 b_3 A_2$ (with $b_3 = c$ if necessary, adjusting A to Ac). If $a_1 \leq a_2$, then $a_1 b_2 a_2 b_3 \equiv a_1 b_3$. Otherwise, $b_1 a_a b_2 a_2 \equiv b_1 a_2$, as shown in Figure 5.7. This gives the equivalence of A (or Ac) to a shorter worm, so w cannot be the shortest word in its equivalence class.

Figure 5.7: Assuming $a_1 \leq a_2$, (a) $a_1b_2a_2b_3 \preceq a_1b_3$ and (b) $a_1b_2a_2b_3 \succeq a_1b_3$. If $a_1 > a_2$, (c) $b_1a_ab_2a_2 \preceq b_1a_2$ and (d) $b_1a_ab_2a_2 \succeq b_1a_2$.



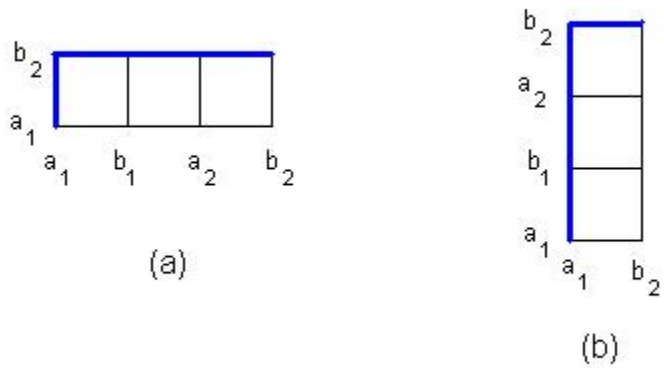
Case 5: There is a local minimum which is not smaller than all previous entries. Locate a local minimum which is larger than both adjacent local minima and label it a_2. If there is none, let a_2 be the last local minimum (and thus the last entry in A). Let $A = A_1a_1b_1a_2b_2A_2$, with $b_2 = c$ if a_2 is the last entry in A. We may assume that $b_1 < b_2$, since it would otherwise have fallen into case 4. But $a_1b_1a_2b_2 \equiv a_1b_2$ (See Figure 5.8), so again w cannot be the shortest word in its equivalence class.

□

Theorem 5.2.2. *Suppose a and b are words of the form in Theorem 5.2.1 with $a \neq b$. Then $a \not\equiv b$.*

Proof. Since $a \neq b$, let i be the first entry where $a_i \neq b_i$ or one of these entries does not exist. If a_i, b_i both exist, we may assume that a is expanding. If one of the entries does not exist, we may assume it is b_i.

Figure 5.8: (a) $a_1b_1a_2b_2 \preceq a_1b_2$ and (b) $a_1b_1a_2b_2 \succeq a_1b_2$.



Case 1: Suppose $a_i > b_i$ and φ is a witness that $a \preceq b$. Since both worms are expanding up to this point, in the scheduling φ, for $j \leq i$ we can note the following:

- For any local maximum a_j, any points (j, k) on the schedule must have $k \geq j$.
- For any local minimum b_j, any points (k, j) on the schedule must have $k \geq j$.

If a_i is a local maximum, we know that the points $(i, i-1), (i, i), (i-1, i+1)$ are unavailable to our schedule φ since these do not satisfy the criterion of a witness. By the above notes, the points (i, j) with $j < i-1$ and $(j, i+1)$ with $j \leq i-1$ cannot be in our schedule. Since the schedule must pass through one of these points, it cannot be a witness.

Similarly, if a_i is a local minimum, the points $(i+1, i_1), (i, i), (i-1, i)$ are

unavailable, so the schedule φ cannot be a witness and $a \not\preceq b$.

Case 2: If $a_i < b_i$, there is no witness of $b \preceq a$. This is similar to Case 1.

Case 3: Suppose b_i does not exist, and a is expanding up to entry i. If a_i is a local maximum, then $a \not\preceq b$ since $a_i > a_j = b_j$ for any $j < i$ (Theorem 5.1.1). If a_i is a local minimum, then $b \not\preceq a$, again by Theorem 5.1.1.

Case 4: If b_i does not exist and a is contracting at entry i, then we look at the last entry in our two words. Suppose b_{i-1} is a local minimum. Then $b_{i-1} = a_{i-1} < a_j$ for $j \geq i$. Specifically, b_{i-1} is smaller than the last entry in a, so $a \not\preceq b$ by Theorem 5.1.1. Similarly, if b_{i-1} is a local maximum, then $b \not\preceq a$ by Theorem 5.1.1.

Thus, $a \not\equiv b$, as desired. □

As indicated in the proofs above, the point where a worm changes from expanding to contracting is very important. Unfortunately, it isn't well-defined whether a certain index is expanding or contracting, and some worms (such as $a = 0, 1$) are both expanding and contracting. To avoid this confusion, define the **fulcrum** of a worm to be the average index of the global extrema in the worm. So, at indices less than or equal to the fulcrum, the worm is expanding, and at indices larger than or equal to the fulcrum, it is contracting.

The index of the fulcrum falls into one of three categories. For worms of length 0 (for example, $a = a_0 = 2$), the fulcrum is 0 since the 1 entry is both the global maximum and the global minimum. For worms which have a unique global maximum

and a unique global minimum, these two indices are adjacent, say at i and $i+1$, so the fulcrum is $i+\frac{1}{2}$. For example, the worm $a = 0, 2, -1, 1$ has a fulcrum of $\frac{3}{2}$. The remaining worms will have a repeated global maximum, or a repeated global minimum. If the indices of the repeated extrema are $i-1$ and $i+1$, the remaining extremum is at index i, which is also the fulcrum. For example, $a = 0, 1, 0$ has a fulcrum of 1.

Consistent with this definition, the fulcrum of an expanding worm of length m is $m - \frac{1}{2}$ and the fulcrum of a contracting worm is $\frac{1}{2}$, no matter the length. This becomes useful for finding an algorithm for the sum of two worms, covered in the next section.

5.3 The Sum Worm

For two worms a and b, we wish to form a sum worm. Since two worms may have differing lengths, we cannot just add the entries, so we again look at schedules.

Definition 5.3.1. *Let a, b be worms and φ a schedule of a versus b. The word $c = a +^{\varphi} b$ is defined as having $i + j$ entry $a_i + b_j$ if the point (i, j) is on the schedule φ. The* **sum worm***, $a + b$, is the worm associated with $a +^{\theta} b$ where θ is the schedule so that $a +^{\theta} b \preceq a +^{\varphi} b$ for any schedule φ.*

Theorem 5.3.1. *$a + b$ is well-defined.*

Proof. Let m be the length of a, n the length of b, f_a be the fulcrum of a and f_b the fulcrum of b. Construct the schedule θ, beginning at $(0, 0)$ and proceeding from a point (i, j) by the first applicable case in the following algorithm, where if the

referenced entries do not exist, they are considered to have the same value as the final entry of the respective worm:

0. If $i = m$ and $j = n$, then this is the end of the schedule.

1. If $i = m$, then proceed NORTH to $(i, j+1)$.

2. If $j = n$, then proceed EAST to $(i+1, j)$.

3. If $a_i > a_{i+1}$, proceed EAST to $(i+1, j)$.

4. If $b_j > b_{j+1}$, proceed NORTH to $(i, j+1)$.

5. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} - a_i < b_{j+1} - b_j$, proceed EAST to $(i+1, j)$.

6. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} - a_i > b_{j+1} - b_j$, proceed NORTH to $(i, j+1)$.

7. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} - a_i = b_{j+1} - b_j$, proceed NORTH or EAST.

8. If $i \leq f_a - 1$ proceed EAST to $(i+1, j)$.

9. If $j \leq f_b - 1$ proceed NORTH to $(i, j+1)$.

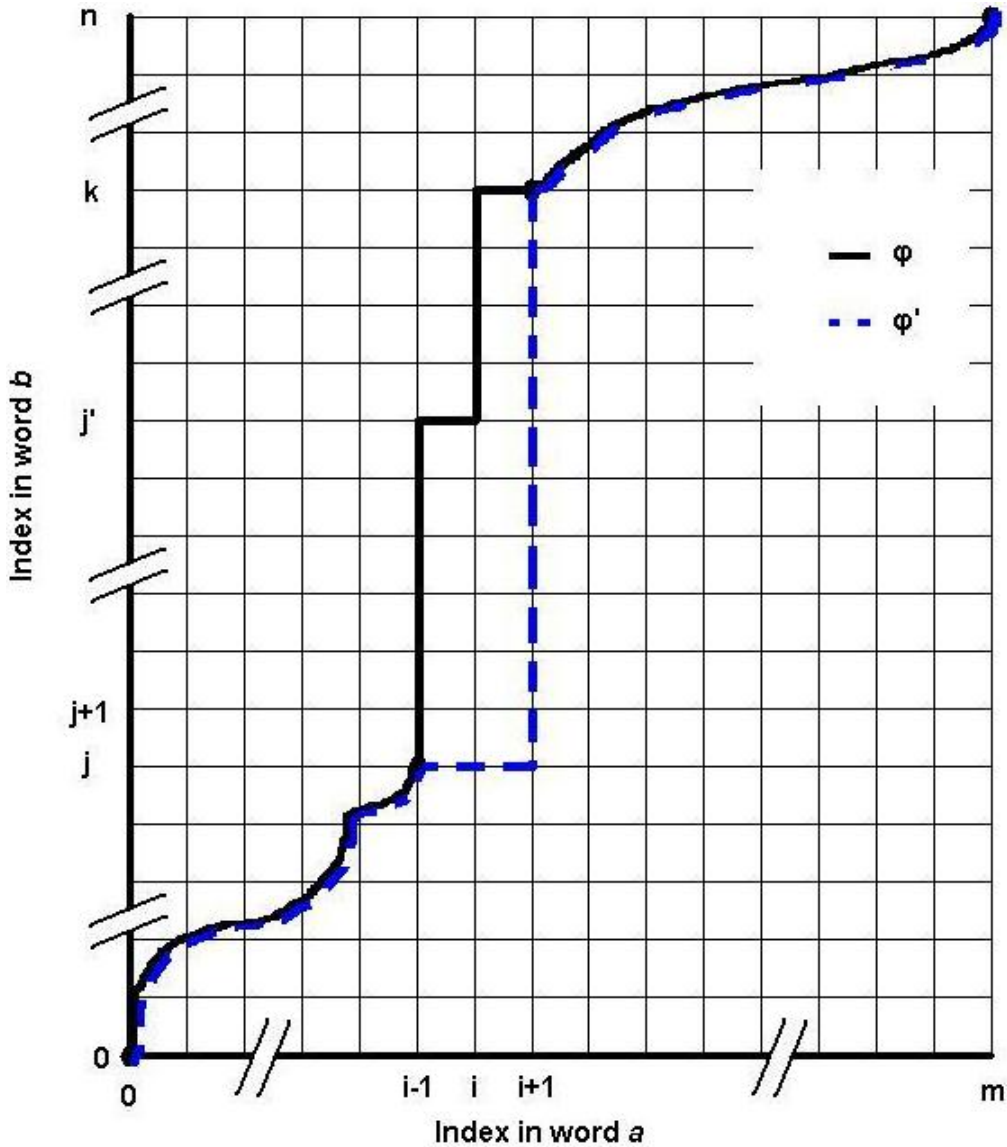
10. If $a_{i+1} - a_{i+2} > b_{j+1} - b_{j+2}$, proceed EAST to $(i+1, j)$.

11. If $a_{i+1} - a_{i+2} = b_{j+1} - b_{j+2}$, proceed NORTH or EAST.

12. Otherwise, proceed NORTH to $(i, j+1)$.

This constructed $a +^{\theta} b$ is the desired minimal sum. Of all the schedules which have a sum word which is not preceded by $a +^{\theta} b$, let φ be the one which agrees with θ for the longest time. Without loss of generality, we may assume that θ travels east

Figure 5.9: Construction of the schedule φ'



and φ travels north when they split, and let i+1 be the index of the first minimum in a after the split, j the index of b at the split, and k the first index where $(i{+}1, k)$ is on φ. For future reference, the number j' is the first index where (i, j) is on φ. Note that

$j' = j$ is possible. Construct a schedule φ' by following both schedules to the split, then travelling east to $(i+1, j)$, which is on θ by our construction, then travelling north to $(i{+}1, k)$, and finally following φ to its end. This is shown in Figure 5.9. From the definitions, we can write $a +^{\varphi'} b = AB'C$ and $a +^{\varphi} b = ABC$ where the two words differ only in B and B'. With this definition, $B = a_{i-1}{+}b_j \cdots a_{i-1}{+}b_{j'}\, a_i{+}b_{j'} \cdots a_i{+}b_k$ and $B' = a_{i-1}{+}b_j\, a_i{+}b_j\, a_{i+1}{+}b_j\, a_{i+1}{+}b_{j+1} \cdots a_{i+1}{+}b_k$.

Supposing that $a +^{\varphi'} b \preceq a +^{\varphi} b$, we know that $a +^{\theta} b \preceq a +^{\varphi'} b$ since φ' agrees with θ for longer than φ. But transitivity would say that $a +^{\theta} b \preceq a +^{\varphi} b$, a contradiction, so $a +^{\theta} b$ must be minimal. It only remains to be shown that $a +^{\varphi'} b \preceq a +^{\varphi} b$. This falls into three cases:

Case 1: $i \leq f_a$ and $j \leq f_b$. To demonstrate $B' \preceq B$, we use the following schedule. Here, the labels of the indices are used rather than the indices themselves to help avoid confusion. We must begin at the point with x and y labels $a_{i-1}{+}b_j$. Travel north to the point with y label $a_{i-1} + b_{i+1}$, one step. Travel east for two steps to the point with x label $a_{i+1} + b_j$. Begin alternating two steps north with two steps east, so that the stopping points along the y-axis are the maximums and the stopping points along the x-axis are the minimums, until reaching $a_{i-1}{+}b_{j'}$ in y. In so doing, we compare $a_{i-1}{+}b_l$ to $a_{i+1}{+}b_l$. Since a is expanding, $a_{i-1}{+}b_l \geq a_{i+1}{+}b_l$ and the choice of path maintains the comparison at the intervening steps. Take one step north, advancing from $a_{i-1}{+}b_{j'}$ to $a_i{+}b_{j'}$ in y. If $j = j'$, this northward step is where we begin our schedule. Continue with alternating two steps north and two steps east until reaching $a_i{+}b_k$ in y, and take the last northward step to finish the schedule. See Figure 5.10 for

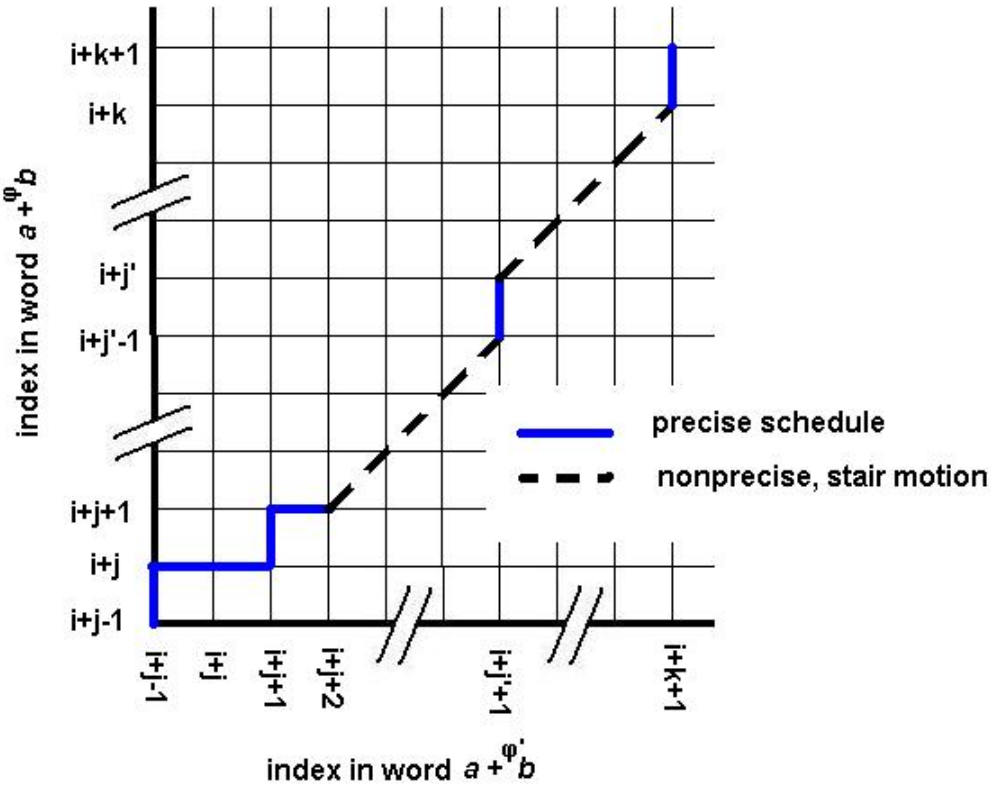


Figure 5.10: A witness for Theorem 5.3.1, case 1.

clarification.

Case 2: $i \leq f_a$ and $j > f_b - 1$. (Note that if $i > f_a$, then $j > f_b$ by the construction of θ.) To demonstrate $B' \preceq B$, we use the following schedule. Here, the labels of the indices are used rather than the indices themselves to help avoid confusion. We must begin at the point with x and y labels $a_{i-1}+b_j$. Begin by travelling north to the y point with label $a_j+b_{j'}$. Note that since b is contracting and j is a minimum, b_j is the smallest b point, and each of these moves is allowed. Then

move east to the point with label $a_{i+1}+b_{j'}$. Note that since b is contracting, we need only check that $a_{i+1}+b_{j+1} < a_i+b_{j'}$. But $b_{j+1}-b_{j'} < b_{j+1}-b_j < a_j-a_{i+1}$. Proceed with two steps north then two steps east, as in Case 1. The final step should again be northward.

Case 3: $i > f_a$ and $j > f_b$. To demonstrate $B' \preceq B$, we use the same schedule as in Case 2. To verify $a_{i+1}+b_{j+1} < a_i+b_{j'}$ recall that $a_i-a_{i+1} > b_{j+1}-b_{j+2} > b_{j+1}-b_{j'}$.

Thus $a +^{\varphi'} b \preceq a +^{\varphi} b$, concluding the proof. □

5.4 The Set of Worms Forms a Lattice

As we look at worms and the partial order on them, we may ask what properties this partial order exhibits. In fact, it forms a lattice, with a meet and a join. In order to show this, we give a definition for something called a join, show it is well-defined, and then show it is what we would like it to be, specifically that if $a \preceq c$ and $b \preceq c$ then $a \vee b \preceq c$. We define the meet similarly, but the proof that it is well-defined and what would be expected of a meet are omitted as they follow this proof of the join exactly.

Definition 5.4.1. *Let a, b be worms and φ a schedule of a versus b. The word $c = a \vee^{\varphi} b$ has $i + j$ entry $\max\{a_i, b_j\}$ for every point (i, j) on the schedule φ. The* **join** *$a \vee b$ is the worm associated with $a \vee^{\theta} b$ where θ is the schedule so that $a \vee^{\theta} b \preceq a \vee^{\varphi} b$ for any schedule φ. Similarly, define $a \wedge^{\varphi} b$ as the word with $i + j$ entry $\min\{a_i, b_j\}$ for every point (i, j) on the schedule φ and the* **meet** *$a \wedge b$ as the*

worm associated with $a \wedge^{\theta} b$ where θ is the schedule so that $a \wedge^{\theta} b \preceq a \wedge^{\varphi} b$ for any schedule φ.

Theorem 5.4.1. *$a \vee b$ is well-defined.*

Proof. Let m be the length of a, n the length of b, f_a be the fulcrum of a and f_b the fulcrum of b. Construct the schedule θ, beginning at $(0, 0)$ and proceeding from a point (i, j) by the first applicable case in the following algorithm:

0. If $i = m$ and $j = n$, then this is the end of the schedule.
1. If $i = m$, then proceed NORTH to $(i, j+1)$.
2. If $j = n$, then proceed EAST to $(i+1, j)$.
3. If $a_i > a_{i+1}$, proceed EAST to $(i+1, j)$.
4. If $b_j > b_{j+1}$, proceed NORTH to $(i, j+1)$.
5. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} < b_{j+1}$, proceed EAST to $(i+1, j)$.
6. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} > b_{j+1}$, proceed NORTH to $(i, j+1)$.
7. If $i \leq f_a - 1$, $j \leq f_b - 1$ and $a_{i+1} = b_{j+1}$, proceed NORTH or EAST.
8. If $i = m - 1$, proceed NORTH to $(i, j+1)$.
9. If $j = n - 1$, proceed EAST to $(i+1, j)$.
10. If $a_{i+1} > b_{j+1}$, proceed EAST to $(i+1, j)$.
11. If $a_{i+1} = b_{j+1}$, proceed NORTH or EAST.

12. Otherwise, proceed NORTH to $(i, j+1)$.

This constructed θ is the desired minimal element. Of all the schedules which have a sum word which is not preceded by $a \vee^{\theta} b$, let φ be the one which agrees with θ for the longest time. Without loss of generality, we may assume that θ travels east and φ travels north when they split, and let $i+1$ be the index of the first minimum in a after the split, j the index of b at the split, and k the first index where $(i+1, k)$ is on φ. For future reference, the number j' is the first index where (i, j') is on φ. Note that $j' = j$ is possible. Construct a schedule φ' as in the proof of Theorem 5.3.1 and, for each of the cases, let the schedule of $a \vee^{\varphi'} b$ versus $a \vee^{\varphi} b$ be the same as the schedule of $a +^{\varphi'} b$ versus $a +^{\varphi} b$ in the corresponding case of the previous proof. It only remains to show that these schedulings witness $a \vee^{\varphi'} b \preceq a \vee^{\varphi} b$.

Case 1: Since a is expanding, $\max\{a_{i-1}, b_l\} \geq \max\{a_{i+1}, b_l\}$.

Case 2: We need only check that $\max\{a_{i+1}, b_{j+1}\} < \max\{a_i, b_{j'}\}$. b_{j+1} must be a maximum in b, and $j > f_b - 1$ gives that $a_i > b_{j+1}$. Since $a_{i+1} < a_i$ (a_i is a maximum a), the inequality is shown.

Case 3: Again, we need only show that $\max\{a_{i+1}, b_{j+1}\} < \max\{a_i, b_{j'}\}$. Since the proof in Case 2 relies only of $j > f_b - 1$ and not $i \leq f_a$, this is simply a repetition of the previous argument.

Therefore, $a \vee^{\varphi'} b \preceq a \vee^{\varphi} b$, as desired.

□

Theorem 5.4.2. *If $a \preceq c$ and $b \preceq c$ then $a \vee b \preceq c$.*

Proof. Suppose φ witnesses $a \preceq c$ and ψ witnesses $b \preceq c$. Construct a schedule ϑ of a versus b by starting at $(0,0)$ with $k = 0$. At a point (i,j) we have a value k. If $(i+1,k)$ is on the path φ, proceed to $(i+1,j)$ on ϑ. Otherwise, if $(j+1,k)$ is on the path ψ, proceed to $(i,j+1)$ on ϑ. Otherwise, increment k.

Consider the word $a \vee^{\vartheta} b$. A point (i,j) on ϑ implies by the construction of ϑ that there is a k so that (i,k) is on φ and (j,k) is on ψ. Therefore, we can define a schedule θ so that $(i+j,k)$ is on θ whenever (i,j) is on ϑ and for every k satisfying the above condition. Since $a_i \leq c_k$ and $b_j \leq c_k$ by the choice of φ and ψ, respectively, the $i+j$ entry of $a \vee^{\vartheta} b$, $\max\{a_i, b_j\}$, is less than c_k so θ is a witness that $a \vee^{\vartheta} b \preceq c$. Thus, by Theorem 5.4.1, $a \vee b \preceq c$. $\square$

Theorem 5.4.3. *$a \wedge b$ is well-defined. In addition, if $c \preceq a$ and $c \preceq b$ then $c \preceq a \wedge b$.*

These theorems combine to show that the set of worms forms a lattice under worm order.

5.5 Counting Worms

Define $w(n)$ to be the number of worms on the alphabet $[n]$. We say a worm has range S if every element in S appears in the worm and all the entries in the worm are represented in S.

Theorem 5.5.1.

$$w(n) = \frac{1}{9}\left(4^{n+1} - 3n - 4\right)$$

Proof. Suppose $w = a_1b_1a_2b_2\cdots a_mb_m$ is any expanding worm, where the a_is and b_is are the entries, with $a_1 = \epsilon$ if w has an odd number of entries. Define $s(w) = b_1a_1b_2a_2\cdots b_ma_m$. This is also an expanding worm, and it has the same range as w. Also, $s(s(w)) = w$ so r is a bijection among expanding worms.

Extend this definition to contracting worms by defining $s(w) = \mathrm{rev}(s(\mathrm{rev}(w)))$ when w is contracting.

Every worm with range $[n]$ has one of four forms, defined by the location of the entries 0 and $n-1$:

$$\begin{aligned} w_1 &= s(a)\,(n-1)\,(0)\,b \\ w_2 &= a\,(0)\,(n-1)\,s(b) \\ w_3 &= s(a)\,(n-1)\,(0)\,(n-1)\,s(b) \\ w_4 &= a\,(0)\,(n-1)\,(0)\,b \end{aligned}$$

where $w = a(0)b$ is a worm with range $\{0, 1, \ldots, n-2\}$ and $a, \mathrm{rev}(b)$ are arbitrary expanding worms on the alphabet $\{1, \ldots, n-2\}$.

Define $v(n)$ to be the number of worms with range exactly $\{0, 1, \ldots, n-1\}$ and $u(n)$ as the number of worms with range exactly $[n]$ where the entry 0 appears twice. We can get these numbers recursively as

$$v(n) = 4(v(n-1) - u(n-1)) \tag{5.5.1}$$

$$u(n) = v(n-1) - u(n-1). \tag{5.5.2}$$

Unwinding these, with the initial counts $v(1) = 1, u(1) = 0$ gives $v(n) = 4 \cdot 3^{n-2}$ and $u(n) = 3^{n-2}$ for $n \geq 2$. Combining this,

$$w(n) = \sum_{k=1}^{n} \binom{n}{k} v(k) = n + 4 \sum_{k=2}^{n} \binom{n}{k} 3^{k-2} = \frac{1}{9}\left(4^{n+1} - 3n - 4\right).$$

□

Alternative proof of $v(n) = 4 \cdot 3^{n-2}$. Suppose that we have an expanding worm $w = w_0 w_1 \ldots w_{n-1}$ with range $[n]$ and $w_{n-1} = 0$. The entry $n-1$ must appear somewhere, and it must be a maximum in the alternating sequence. But no larger maximums are available, and w_{n-2} must be the largest maximum, so $w_{n-2} = n - 1$. Similarly, $w_{n-3} = 1$, $w_{n-4} = n - 2$, and so on. In all, there is only one worm w satisfying these conditions.

Similarly, if we select a worm w on the alphabet $[n]$, it is one of the four types above (w_1, w_2, w_3 and w_4) and has two associated sets A and B, the ranges of a and b, respectively. Given the two sets and the type of the worm, the worm is unique.

For a worm w with range $[n]$, $A \cup B \cup \{0, n-1\} = [n]$. The entries $0, n-1$ cannot occur in a or b, but every other $i \in [n]$ must occur in one of them, so there are three choices for this entry: $i \in A\ B$, $i \in B\ A$, or $i \in A \cap B$. To select a worm, we select one of the four types, then select one of the three sets that each remaining entry can be in, and the worm is uniquely determined. Thus, there are $4 \cdot 3^{n-2}$ worms with range $[n]$. □

5.6 Worms and Percolation

In Section 4.3, we noted that if a_i, b_j are random with uniform distribution on $[t, 1]$ for $0 \leq i < m, 0 \leq j < n$, then the probability $P(m, n)$ is equal to the probability for words $a = a_0 a_1 a_2 \ldots a_{m-1} 0$ and $c = c_0 c_1 c_2 \ldots c_{n-1} 1$ that $a \preceq c$ in worm order. If a and c were worms, each instance would be equivalent to one of the $\binom{m+n}{m}$ different permutations (we know the final character of a is the smallest and the final character of c is the largest, so it remains only to pick which worm the remaining characters lie in), but this is not the case. It is true that a and c are words, but they are usually not worms.

Consider only a. We say that bf a has the same order type as $\sigma \in S_m$ if

$$a_{\sigma(1)-1} < a_{\sigma(2)-1} < a_{\sigma(3)-1} < \cdots < a_{\sigma(m)-1}.$$

Given a specific $\sigma \in S_m$, the probability that a random word a has the same order type as σ is $1/m!$. We wish to compress the word $\sigma(0) = \sigma(1)\sigma(2)\sigma(3)\ldots\sigma(m)$ into an expanding worm since the trailing 0 implies that $\sigma(0)$ is of type w_4 in the proof of Theorem 5.5.1 with $b = \epsilon$. Thus, there are 2^{m-1} possible worms it can compress into since every $1 \leq i < m$ may or may not appear (but appears only once, if it does) and m must appear.

Suppose $w \equiv a_0 a_2 \ldots a_{m-1} 0$ is a worm on the alphabet $[t, 1] \cup 0$ and $w = w_0 w_1 \ldots w_{2k} w_{2k+1} 0$ where $w_0 = \epsilon$ if $a_0 > a_1$. Suppose, in addition, that the process in Section 3.5 assigns the point $\varphi(k) = (i_0, j_0)$ as the last point on the path φ before iterating when run on an instance $(a, b) \in \Omega_2$ with $a_m < t$. Let $c = 1 + t - b$

and assume $b_j > t$ for $j < M$ where M is really big (in other words, with high probability percolation will only occur if there is an open path to the line $x = m$). Suppose also that $w_{i_1} = a_{i_0}$. (While this condition is satisfied for many of the i_0 which are assigned last, it is not satisfied for all of them.) Then the process will reach a point on the line $x = i_2$ with $a_{i_2} = w_{i_1+2}$ if and only if there is a $c_{j_1} > w_{i_1+1}$ before a $c_{j_2} < w_{i_1}$. This is due to the fact that the point (i_0, j_2) is closed and a_{i_0} is the lowest value with $x < i_3$ where $a_{i_3} = w_{i_1+1}$. If there is not a $c_{j_1} > w_{i_1+1}$ with $j_1 < j_2$, all the points (i_3, j) with $j \leq j_2$ are closed, and the instance does not percolate.

Given k, define the event E_i with $0 \leq i \leq k$ as the event that, for uniformly random instance $(a, b) \in \Omega_2$ satisfying all the conditions above, there is a $c_j > w_{2i+1}$ before any c_k with $c_k < w_{2i}$. If $w_0 = \epsilon$, define E_0 as the event that $c_0 > a_0 = w_1$. Then the events E_i are non-negatively correlated, which is to say

$$\Pr[E_i \wedge E_j] \geq \Pr[E_i] \Pr[E_j].$$

Thus, for a given k, the probability of percolation is $\Pr[E_0 \wedge E_1 \wedge \cdots \wedge E_k]$. Therefore, the probability of percolating for a given k is at least 2^{-k}.

This would give a bound on $\theta(t)$ if one could compute $\Pr[a \equiv w | l(w) = 2(k+1)]$ where a is overloaded to mean the word $a_0 a_1 \ldots a_{m-1} 0$ as well as the uniformly random function $\mathbb{N} \to [0, 1]$ with $a_i \in [t, 1]$ for $i < m$ and $a_m < t$ and w is defined above. However, efforts to compute this probability have not been successful.

Chapter 6

A Combined Approach

Recall from Chapter 4 that we can associate a permutation $\sigma \in S_{m+n}$ to a configuration $p_t(\sigma) \in T_t(m,n)$ specifying whether the vertices of $[m] \times [n]$ are open or closed. In Chapter 3, we found an algorithm to determine whether or not a configuration percolates based on reading successive values horizontally until reaching a closed point and then reading successive values vertically until finding a better value or another closed point. There are drawbacks to both approaches. Although the correspondence between instances and permutations is predictable, the correspondence between configurations and permutations is not. The algorithm is Section 3.3 is very robust, and can be altered so that it outputs a sequence $(x_0, x_1, x_2, \ldots, x_n)$ for some n, and we know whether or not this percolates. However, the number of such sequences is still very large, so it's hard to say what the probability of percolation is in general.

This chapter focuses on combining these approaches. From a given instance

$\sigma \in \Omega_2$ we will associate a single permutation $\tau \in S_n$. This is done in Section 6.1. In Section 6.2, we distill the association so it is easier to analyze, resulting in a formula for the probability of percolation in Section 6.3.

6.1 Associating Instances and Permutations

Define a function $\varpi_t \colon \Omega_2 \to \Omega_2$ by $\varpi_t(a, b) = (a, \pi_t(b))$ where the new function $\pi_t(b)$ is defined by

$$\pi_t(b)(n) = \begin{cases} 1 + t - b(n) & b(n) \geq t \\ b(n) & b(n) < t. \end{cases}$$

Notice that this is an involution of Ω_2 so choosing a uniformly random (a, b) is the same as choosing a uniformly random $(a, \pi_t(b))$. This changes the percolation map Θ_t, however. We now define a new map $\zeta_t \colon \Omega_2 \to \mathcal{F}_2$ by $\zeta(\sigma)(i, j) = 1 \Leftrightarrow a_i < t, c_j < t,$ or $a_i < c_j$ where $\sigma = (a, c) \in \Omega_2$. Now $\overline{\Theta_t}(\sigma) = 1$ if and only if 0_n is in an infinite open cluster in $\zeta_t(\sigma)$. Notice that the following diagram commutes, so this does not change the probability function θ_t:

$$\begin{tikzcd}
 & \mathcal{F}_2 \arrow[dd, dotted] \\
\Omega_2 \arrow[ur, "f_t"] \arrow[dr, "\Theta_t"] \arrow[d, bend right, "\varpi_t"'] & \\
 & \{0,1\} \\
\Omega_2 \arrow[u, bend right, "\varpi_t"'] \arrow[ur, "\overline{\Theta_t}"] \arrow[dr, "\zeta_t"] & \\
 & \mathcal{F}_2 \arrow[uu, dotted]
\end{tikzcd}$$

The process described in Section 3.5 is restated here with the necessary changes for an instance $(a, c) \in \Omega_2$. This process, like the previous one, keeps track of the best valued point while looking at successive values of a_i until it finds a closed point, then it looks at successive values of c_j, until finding one which is high enough that the point on the same line as the closed point it previously found is open or finding another closed point. The process is repeated until it finds closed points in both directions or an open line. Begin with $\varphi(0) = (0, 0)$ and $k = 0$. While $\varphi(k) = (i_0, j_0)$, increment i, beginning with i_0, until locating an i so that

1. $a_i < t$ — the configuration percolates.

2. $t < a_i < a_{i_0}$ — repeat the process with the segment between (i_0, j_0) and (i, j_0) appended to φ, incrementing k so that $\varphi(k) = (i, j_0)$.

3. $a_i > c_{j_0}$ — increment j, beginning with j_0, until locating a j so that

   (a) $c_j < t$ — the configuration percolates.

   (b) $c_j > a_i$ — repeat the process with the segment between (i_0, j_0) and (i_0, j) appended to φ, incrementing k so that $\varphi(k) = (i_0, j)$.

   (c) $t < c_j < a_{i_0}$ — the configuration does not percolate.

Since this process results in the same path for (a, c) as the process in Section 3.5 for $\varpi_t(a, c)$, the proof is omitted.

For this process and all of the processes so far, we have fixed an instance $(a, b) \in \Omega_n$ and examined its properties. However, for a uniformly random $(a, b) \in \Omega_n$, all the values a_i, b_j are independent and uniform on $[0, 1]$. Thus, if we fix any values,

such as a_i for $i \leq i_0$ and b_j for $j \leq j_0$, the remaining values are still independent and uniform on $[0, 1]$. This observation results in an equivalence between processes on a fixed instance, such as we have used so far, and processes which generate the values as needed, as in the process below. This process keeps track of the minimum and maximum values, μ and ν, respectively, and generates values in the first direction until finding one that is higher than ν, indicating a closed point, then generates values in the second direction until finding one higher than the previous high, which indicates an open point on this line, or one lower than μ, indicating a closed point. One observes that the probability of success in this process is the same as the probability of percolation in the previous process.

Begin by generating X, Y each uniform on $[0, 1]$. If $Y < t$ or $X < t$, succeed. If $Y < X$, fail. Otherwise, set $\mu = X$ and $\nu = Y$. Regenerate X multiple times until it falls into one of the below ranges:

1. $X < t$ — succeed.
2. $t < X < \mu$ — iterate with $\mu = X$.
3. $\nu < X$ — set $\nu = X$ and regenerate Y multiple times until it falls into one of the below ranges:
   (a) $Y < t$ — succeed.
   (b) $t < Y < \mu$ — fail.
   (c) $\nu < Y$ — iterate with $\nu = Y$.

Notice that in this process we succeed whenever we generate a value of X or Y

which is less than t, since this indicates an open line. At any time, the probability that the next value is less than t is exactly t. Thus, if the process fails, it must fail before the first value less than t is reached. Therefore, the process is equivalent to a process wherein it is first decided which value will be the first less than t, and the values before this are generated uniformly on $[t, 1]$. The probability that the first value less than t is the n+1st value generated is given by a geometric distribution, so it is $t(1-t)^n$.

The process in Table 6.1 is the same as the previous processes, although presented in a different format. The idea is that the generated values are in the first direction until one is found that is higher than all the values in the other direction (as in choice 3 above). This generation is on the left side of the table. Once this is found, values are in the second direction until one is found that is lower than all the values in the other direction (as in 3b above) or one that is higher than the bad value from the first direction (as in 3c above). This generation is on the right side of the table. Again, μ represents the minimum, ν the maximum, and k the number of values generated.

Notice that the probability of success in all these processes is the same, which results in the following theorem:

Theorem 6.1.1. *The probability $\theta(t)$ of percolation for a uniformly random $\sigma \in \Omega_2$ can be calculated by*

$$\theta(t) = \sum_{n=0}^{\infty} Q_n t(1-t)^n$$

where Q_n is the probability of success of the process in Table 6.1 when the geometric variable N is n.

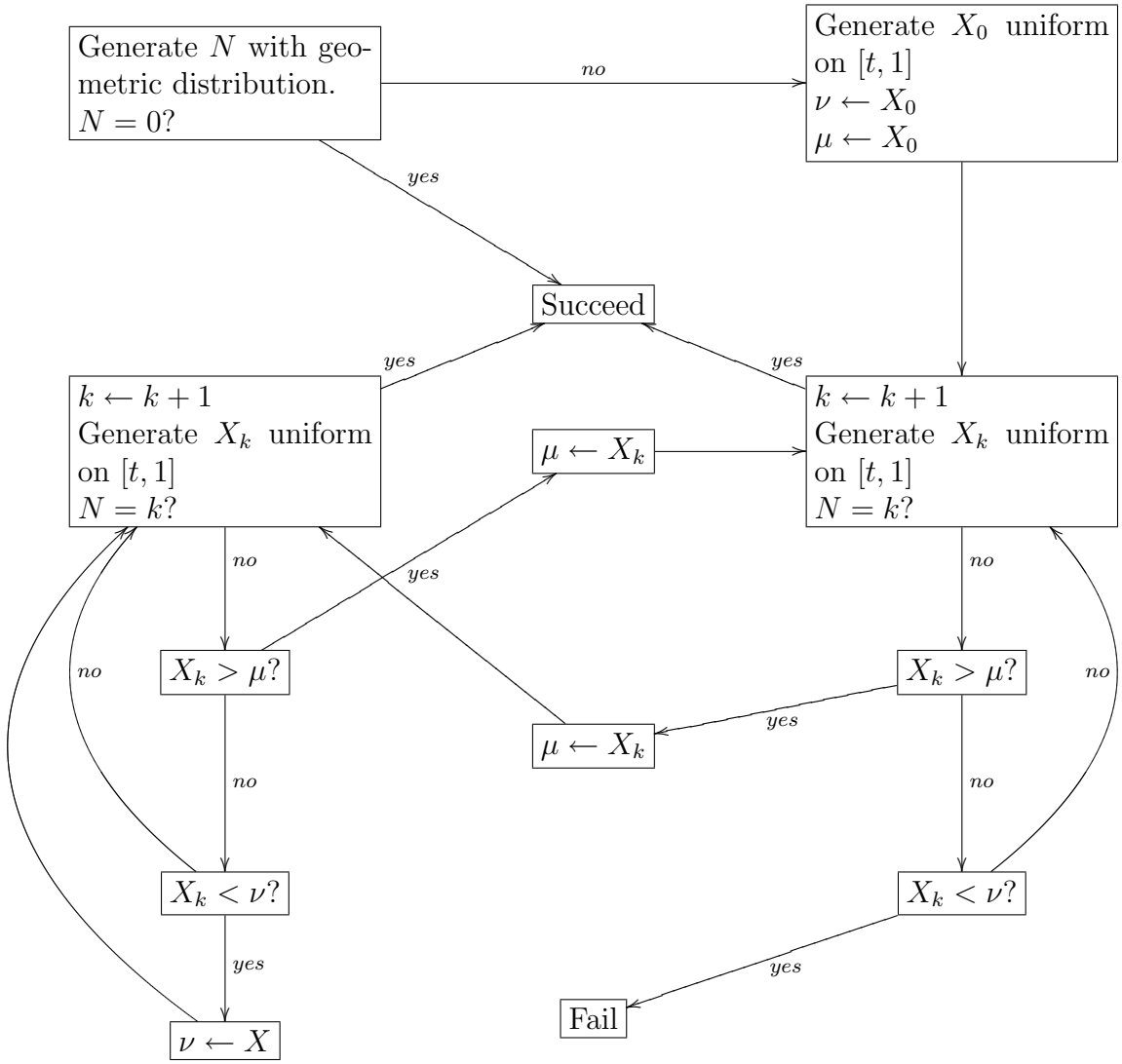


Table 6.1: A process with probability of success $\theta(t)$.

This reduces the problem of calculating $\theta(t)$ to calculating Q_n for $n \geq 0$. Notice that $Q_0 = Q_1 = 1$. There are two processes that would help to compute Q_n. Since N is determined, we eliminate the random variable N in the process in Table 6.1 by conditioning that $n > k$, where k is the current time in the process. This process

is one of the two we could examine in order to calculate Q_n. The other process is described in Table 6.2 and described in the next section.

6.2 A Stochastic Process

Recall from Section 4.1 that for any $\sigma \in S_n$ and independent, identically distributed continuous random variables $X_1, X_2, \ldots, X_n$ the probability that $X_{\sigma(1)} < X_{\sigma(2)} < \cdots < X_{\sigma(n)}$ is $1/n!$. The process in Table 6.1 generates X_k, an independent, uniform random variable on $[t, 1]$, which falls somewhere in the order of $X_0, X_1, \ldots, X_{k-1}$, which has already been generated. By this observation, generating X_k is equivalent to adding an element to the permutation σ defined by

$$X_{\sigma(1)-1} < X_{\sigma(2)-1} < X_{\sigma(3)-1} < \cdots < X_{\sigma(k)-1}.$$

Thus, to generate a random permutation uniformly, we assume we have a uniform permutation $\sigma \in S_k$ and extend it to a permutation $\delta(\sigma) \in S_{k+1}$. Given a permutation σ, we consider the bijective map $\delta_n \colon S_n \times [n+1] \to S_{n+1}$ where $\delta_n(\sigma, i) = \sigma'$ and σ' is defined as the mapping

$$\sigma'(j) = \begin{cases} \sigma(j) & j \neq n+1, \sigma(j) \leq i \\ \sigma(j) + 1 & j \neq n+1, \sigma(j) > i \\ i + 1 & j = n+1. \end{cases}$$

Since this mapping is bijective, selecting a uniform $(\sigma, i) \in S_n \times [n+1]$ is equivalent to selecting a uniform $\sigma \in S_{n+1}$. Define

$$\Xi = \bigcup_{n=1}^{\infty} S_n \times [n+1]$$

and $\delta \colon \Xi \to \Xi$ by $\delta(\sigma, i) = \delta_n(\sigma, i)$ where $\sigma \in S_n$. Thus, if the processes in Tables 6.1 and 6.2 were coupled so that the i generated at time k in process 6.2 corresponds to the position in the order of X_k in process 6.1, then process 6.2 would be in State C when X_k is generated on the left in process 6.1, in State A when X_k is generated on the right in process 6.1, and in State D if process 6.1 has failed. Therefore, the probability of success in process 6.1 for a given n is the sum of the probabilities that process 6.2 is in State A and that it is in State C.

At time $k \geq 2$, the probabilities of changing between the states A, C, and D as $k-1$ is incremented to k are given by the transition matrix

$$M_k = \begin{bmatrix} 1 - 1/k & 1/k & 0 \\ 1/k & 1 - 2/k & 1/k \\ 0 & 0 & 1 \end{bmatrix},$$

where the order of the states is A, C, D. For this process, we start in state C at time $k = 2$. This state has the probability vector $(0, 1, 0)$. The eigenvalues for this matrix are $1, \lambda_k = 1 + \frac{1}{k}(\varphi - 2)$ and $\overline{\lambda_k} = 1 + \frac{1}{k}(\overline{\varphi} - 2)$ where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio and $\overline{\varphi} = 1 - \varphi$. These eigenvalues correspond to the eigenvectors $v = (0, 0, 1), v_k = (\varphi, 1, (k-1)/(2 - \varphi - k))$ and $w_k = (\overline{\varphi}, 1, (k-1)/(2 - \overline{\varphi} - k))$.

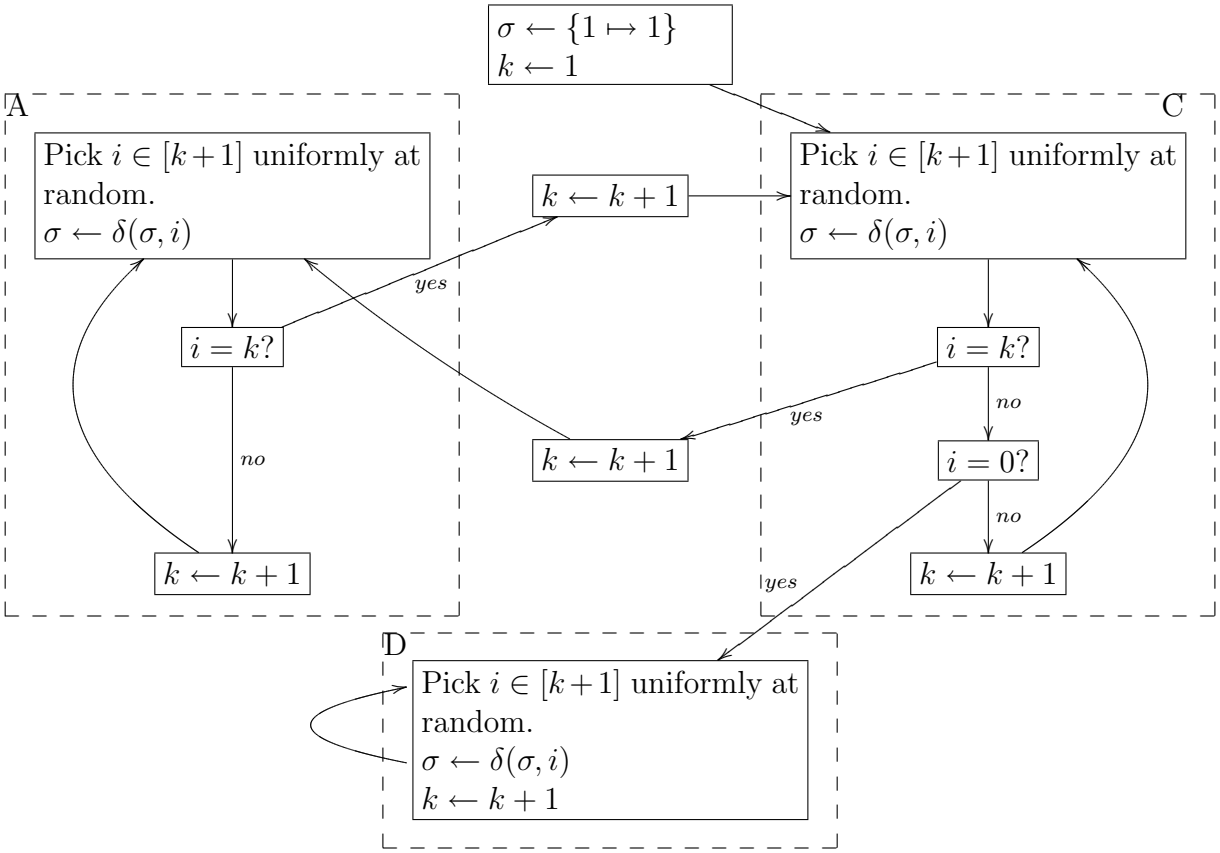


Table 6.2: A process by which we can build random permutations, and decide whether they percolate.

Define

$$\begin{aligned}
\Lambda_k &= \prod_{i=2}^{k} \lambda_i = (-1)^k \binom{1-\varphi}{k} \varphi \\
\overline{\Lambda_k} &= \prod_{i=2}^{k} \overline{\lambda_i} = (-1)^k \binom{\varphi}{k} (-\varphi^{-1})
\end{aligned}$$

where $\binom{r}{k}$ is the generalized binomial coefficient.

Lemma 6.2.1. *If $x = \alpha v_i + \beta w_i + \gamma v$, then $x = \alpha v_j + \beta v_j + \gamma' v$ for some γ'. Consequently, if $x = \alpha v_i + \beta w_i + \gamma v$ then $xM_j = \alpha\lambda_j v_i + \beta\overline{\lambda_j} w_i + \gamma' v$ for some value of γ'.*

Proof. Since $x = (a, b, c) = \alpha v_i + \beta w_i + \gamma v$, we have the system of equations

$$\begin{aligned} a &= \alpha\varphi + \beta\overline{\varphi} \\ b &= \alpha + \beta \\ c &= \alpha\frac{(i-1)}{(2-\varphi-i)} + \beta\frac{i-1}{2-\overline{\varphi}-i} + \gamma. \end{aligned}$$

Notice that the first two equations create a system of two equations and two unknowns. Since $\varphi - \overline{\varphi} \neq 0$, it has a unique solution. To find α', β', γ' where $x = \alpha' v_j + \beta' v_j + \gamma' v$ we solve the system of equations

$$\begin{aligned} a &= \alpha'\varphi + \beta'\overline{\varphi} \\ b &= \alpha' + \beta' \\ c &= \alpha'\frac{(j-1)}{(2-\varphi-j)} + \beta'\frac{j-1}{2-\overline{\varphi}-j} + \gamma'. \end{aligned}$$

The first two equations have the same (unique) solution, so $x = \alpha v_j + \beta v_j + \gamma' v$, as desired.

For the second part,

$$\begin{aligned} xM_j &= (\alpha v_i + \beta w_i + \gamma v)M_j \\ &= (\alpha v_j + \beta w_j + \gamma' v)M_j \\ &= \alpha v_j M_j + \beta w_j M_j + \gamma' v M_j \\ &= \alpha \lambda_j v_j + \beta \overline{\lambda_j} w_j + \gamma'' v \\ &= \alpha \lambda_j v_i + \beta \overline{\lambda_j} w_i + \gamma'' v, \end{aligned}$$

as desired. □

Since we will build any permutation σ in S_n by running this algorithm to time $k = n$, the probability Q_n is the probability that we are not in state D at time n.

6.3 The Percolation Probability

Theorem 6.3.1.

$$\theta(t) = \frac{1+3\varphi}{5} t^{2-\varphi} + \frac{4-3\varphi}{5} t^{1+\varphi}$$

where $\varphi = \dfrac{1+\sqrt{5}}{2}$.[1]

Proof. Define Q_n as the probability that we are not in state D (as in Section 6.2) after time n. Notice that if $N = n$, we can define this to mean that the next line is

[1]Special thanks to Peter Doyle and Mathematica for this closed form.

open (since this always occurs with probability t). So

$$\theta(t) = \sum_{n=0}^{\infty} Q_n t(1-t)^n.$$

However, the stochastic process is not defined unless $n \geq 2$. Consider $n = 0$ and $n = 1$. This corresponds to the configurations where $a_0 < t$ and $b_0 < t$, $(c_0 > 1)$, respectively. Thus we always percolate in these cases. Thus,

$$\theta(t) = t + t(1-t) + \sum_{n \geq 2} Q_n t(1-t)^n.$$

We show now that

$$Q_n = \left(1 - \frac{\varphi}{\sqrt{5}}\right)(\varphi + 1)\Lambda_n + \frac{\varphi}{\sqrt{5}}(2-\varphi)\overline{\Lambda}_n.$$

However, if $w = (a, b, c)$ is our starting probability vector then the final probability vector is $w' = (a', b', c') = wM_2M_3 \cdots M_n$ and the probability that we are not in state D is $Q_n = a' + b' = 1 - c'$. Our starting vector is $w = \alpha v_2 + \beta w_2 + \gamma v$, $wM_2M_3 \cdots M_n = \alpha\Lambda_n v_2 + \beta\overline{\Lambda_n} v_2 + \gamma' v$ for some γ' and $a' = \alpha\Lambda_n\varphi + \beta\overline{\Lambda_n}\overline{\varphi}$, $b' = \alpha\Lambda_n + \beta\overline{\Lambda_n}$ so $Q_n = \alpha\Lambda_n(\varphi + 1) + \beta\overline{\Lambda_n}(\overline{\varphi} + 1)$. For the starting vector $w = (0, 1, 0)$, we have $\alpha = 1 - \frac{\varphi}{\sqrt{5}}$ and $\beta = \frac{\varphi}{\sqrt{5}}$. Since $\overline{\varphi} + 1 = 1 - \varphi + 1 = 2 - \varphi$.

This gives that

$$\begin{aligned}
\theta(t) &= t + t(1-t) + t\sum_{n\geq 2}(1-t)^n \left[\left(1-\frac{\varphi}{\sqrt{5}}\right)(\varphi+1)(-1)^k\binom{1-\varphi}{k}\varphi\right] \\
&\quad + t\sum_{n\geq 2}(1-t)^n\left[\frac{\varphi}{\sqrt{5}}(2-\varphi)(-1)^k\binom{\varphi}{k}(-\varphi^{-1})\right] \\
&= t + t(1-t) \\
&\quad + \frac{3\varphi+1}{5}\left[t^{1-\varphi} - 1 - (1-t)(1-\varphi)\right] \\
&\quad + \frac{4-3\varphi}{5}\left[t^{\varphi} - 1 + (1-t)\varphi\right]
\end{aligned}$$

where the second equality is by the generalized binomial theorem. Algebraic simplification leads to

$$\theta(t) = \frac{1+3\varphi}{5}t^{2-\varphi} + \frac{4-3\varphi}{5}t^{1+\varphi},$$

as desired. □

The simplicity of this formula is astonishing, and further research hopes to discover good combinatorial reasons for this.

Corollary 6.3.1. *The critical exponent,*

$$\beta = \lim_{t\to t_c{}^+}\frac{\ln\theta(t-t_c)}{\ln(t-t_c)},$$

is $2-\varphi$.

Proof. Since $t_c = 0$, $t^{2-\varphi} > t^{1+\varphi}$ and $\ln t < 0$ for $0 < t < 1$,

$$\begin{aligned}
\frac{\ln \theta(t)}{\ln t} &\geq \frac{\ln\left(2\frac{1+3\phi}{5}t^{2-\varphi}\right)}{\ln t} \\
&= \frac{\ln C_1 + (2-\varphi)\ln t}{\ln t} \\
&\geq 2-\varphi
\end{aligned}$$

where $C_1 = 2(1+3\phi)/5$ is a constant. The lower bound is given by

$$\begin{aligned}
\frac{\ln \theta(t)}{\ln t} &\leq \frac{\ln\left(\frac{1+3\phi}{5}t^{2-\varphi}\right)}{\ln t} \\
&= \frac{\ln C_2 + (2-\varphi)\ln t}{\ln t} \\
&\to 2-\varphi \quad \text{as} \quad t \to 0^+
\end{aligned}$$

where $C_2 = (1+3\phi)/5$ is a constant. Therefore,

$$\beta = \lim_{t\to 0^+} \frac{\ln \theta(t)}{\ln(t)} = 2-\varphi,$$

as desired. □

Chapter 7

Generalizations

There are many different generalizations that can be made for coordinate percolation, and some of them are explored here. The first problem presented in Section 7.1 is not a generalization, but a reiteration of the original question asked in Section 2.2. Although we have computed the probability of percolation for the coordinate percolation model in two dimensions, we first asked the generalized question, in n dimensions. Section 7.1 returns to this question and attempts to explore the generalizations of the methods used in computing this probability, specifically the use of the comparison $a_i < c_j$ for open vertices rather than the sum comparison $a_i + b_j < 1 + t$. In addition, this section looks at coordinate percolation on the triangular lattice.

Section 7.2 and Section 7.3 begin to lay out some of the difficulties faced when generalizing coordinate percolation to other distributions, such as the normal distribution, on the random variables $a_i(j)$ and to other operations, such as multiplication, being used to combine the coordinate values in different directions.

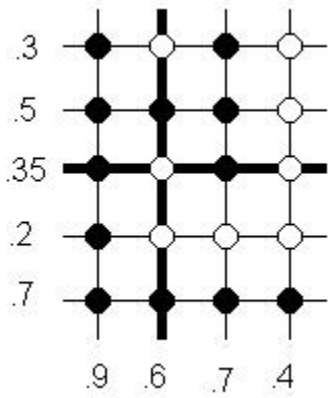


Figure 7.1: A configuration where backtracking may be required.

7.1 Lattice Generalizations

Although we have a percolation probability for the nonnegative integer lattice in two dimensions, there are many other lattices which we may consider. For example, we could consider the entire integer lattice. The nonnegative integer lattice is considered in this paper because we do not need to consider backtracking on this lattice to find open paths containing the origin. However, the proof that backtracking does not need to be considered relies on the fact that on the grid $[n] \times [m]$ we cannot go out of this box. When we include vertices with negative coordinates, two new possibilities arise. The first possibility is that percolation is possible in four different main directions: northeast, northwest, southeast and southwest. There is dependency between the directions, which forces a more thorough evaluation than saying the probability of percolation is four times the probability found in Theorem 6.3.1. The second possibility when vertices with negative coordinates are included is configurations such as that in Figure 7.1, where immediate backtracking is required.

However, such configurations are rare enough that they should not change the probability of percolation. Since the analysis in Theorem 2.3.1 still applies, the critical threshold in this model is still 1.

Conjecture 7.1.1. *Let $\alpha(t)$ be the probability of percolation on the two-dimensional integer lattice $\mathbb{Z}^2$ under the coordinate percolation model. Then*

$$1 - (1 - \theta(t))^2 \leq \alpha(t) \leq 1 - (1 - \theta(t))^4$$

where $\theta(t)$ is given in Theorem 6.3.1.

A lower bound of $1 - (1 - \theta(t))^2$ is hypothesized because the first quadrant and the third quadrant are almost entirely independent (sharing only the lines $x = 0$ and $y = 0$) so we have two independent coordinate percolation processes that may percolate. A configuration fails to percolate if both of these fail to percolate. The upper bound is only conjectured because we do not have four independent processes.

In Section 2.2, the problem statement is generalized to the n-dimensional non-negative integer lattice, and some progress is made toward finding the percolation probability in this model in Section 3.3. More progress on this is hopeful, but other generalizations into the n-dimensional case are possible.

In two dimensions, a one-dimensional subspace (a line) and an $n-1$-dimensional subspace (a hyperplane) are equivalent. We generalize to n dimensions by assigning values to the coordinates, or the hyperplanes. However, it is also possible to assign the values to the lines, in which case even the indexing of these values is not trivial.

Another possibility when generalizing to n-dimensions is to generalize the problem

$$
\begin{array}{c}
a(i) < b(j) < c(k) \\
a(i) < c(k) < b(j) \\
b(j) < a(i) < c(k) \\
b(j) < c(k) < a(i) \\
c(k) < a(i) < b(j) \\
c(k) < b(j) < a(i)
\end{array}
$$

Table 7.1: Possible orders when $n = 3$

that we can solve; namely, percolation in the model where (i, j) is open if $a_i < c_j$. How does this generalize to higher dimensions? For $n = 2$, there were only two orders to consider, $a_i < c_j$ and $c_j < a_i$. It is natural, in this case, to say one order is open and the other is closed. For $n = 3$, there are 6 possible orders, listed in Table 7.1. Which of these orders should be open and which of them should be closed? This seems to be based on personal choice, but two possibilities will be presented here.

In the first, we map a instance $(a, b, c) \in \Omega_3$ to a configuration in $\mathcal{F}_3$ where (i, j, k) is open if and only if $a(i) < b(j) < c(k)$. This causes about 1/6 of the vertices to be open, and does not depend on t, but we can still analyze this case and try to bring a parameter back in later. Attempts to work out algorithms in this case have so far been fruitless, but there is hope that progress will eventually be made. What seems to make this hard is having, in effect, two criteria for whether a vertex is open: $a(i) < b(j)$ and $b(j) < c(k)$. From the first criterion, high values for $b(j)$ seem to be good, but the second criterion makes these same high values bad. However, one approach would be to generate new values of a, b, or c depending on the ordering of the high values of a, b, and c and the low values of a, b, and c.

Table 7.2 shows some possible states for a percolation algorithm based on these criteria. The columns represent different orderings for the minimum values of a, b,

	abc	acb	bac	bca	cab	cba
ABC	A	B	A	A	A	A
ACB	C	B	A	A	A	A
BAC	B	B	A	D	A	D
BCA	B	B	A	D	A	D
CAB	C	D	A	D	D	D
CBA	B	D	A	D	D	D

Table 7.2: Possible states to determine whether a configuration percolates if vertices are open when $a(i) < b(j) < c(k)$.

and c while the rows represent different orderings for the maximum values of a, b, and c. For example, the first state in the first row represents the state where the minimum values of a, b, and c are in the order $a < b < c$ and the maximum values are in the order $A < B < C$. The states indicate which function to generate further values for (the first state in the first row generates further values for the function a) with arrows indicating which states to which this state could transfer. D indicates that percolation can not occur if the algorithm finds itself in this state. One of these occurs in row CAB column acb. Suppose the maxima are $a(i), b(j), c(k)$ and the minima are $a(i'), b(j'), c(k')$. Since $c(k) < a(i)$ every vertex (i, x, y) we've seen is closed. Since $c(k) < b(j)$, every vertex (x, j, y) we've seen is closed. Since $c(k') < b(j')$, every

vertex (x, y, k') we've seen is closed. These three planes form a closed box around the origin so percolation does not occur. The other D states are similar.

This would lead to a probability of percolation as the algorithm in Table 6.2 did, except that the probabilities of changing states are unknown. The state abc, ABC is not a single state; it represents a state where the lowest part of the order is a^+b and the highest part of the order is BC^+. Here, we use a^+b to mean that there are some number of a values at the bottom of the order, and this number is at least one, and then there is a b value, before any c values are seen. The transfer to state BAC occurs with probability $1/k$ where we have k values so far, but the transfer to state BCA occurs with probability i/k where there are i values corresponding to c's at the top of the order.

The second possibility is to say a vertex (i, j, k) is open if $a(i) < b(j) < c(k)$ or $b(j) < c(k) < a(i)$ or $c(k) < a(i) < b(j)$. Again, this does not depend on t, but it has the advantage of half the vertices being open. However, the analysis is complicated and it seems unclear what would block percolation. So far, it seems lines with $a(i) < b(j) < a(i) + \varepsilon$ would be good, but that doesn't tell what would make a plane bad, which is what blocks percolation in the case with sums and thresholds.

Further analogues are available if we look at other lattices. The most prominent regular planar lattice other than the integer lattice is the triangular lattice. For the triangular lattice, we would need to assign three coordinates, one for each direction of lines, and these coordinates would be subject to a constraint, since the triangular lattice is two dimensional and therefore cannot have three degrees of freedom. This would allow the assigning of values to all the lines, and the introduction of a threshold

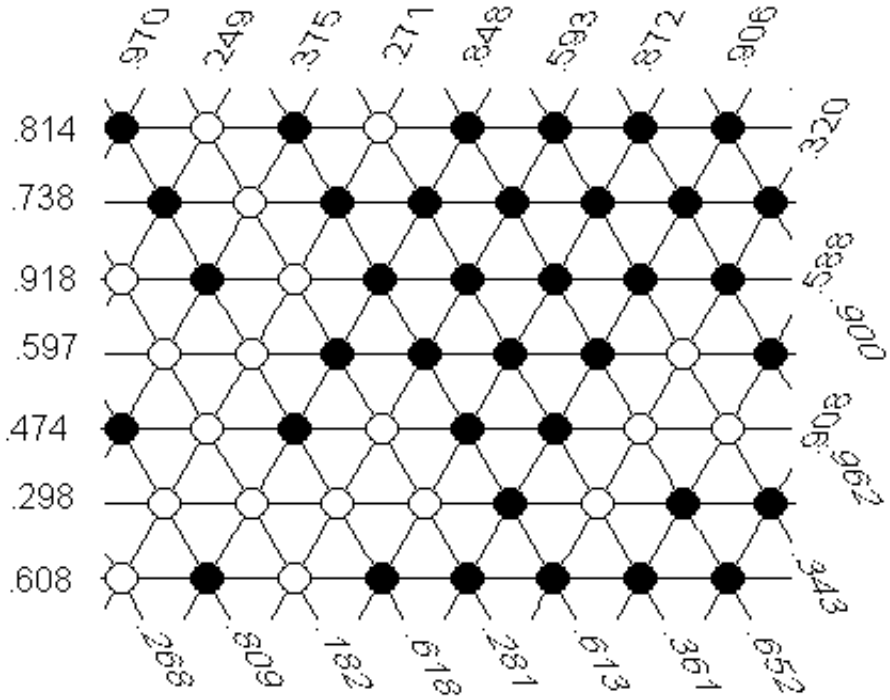


Figure 7.2: Part of a configuration for coordinate percolation on the triangular lattice with threshold 3/2

would indicate which vertices are open and which are closed. One example, with a threshold of 3/2, given in Figure 7.2.

With a threshold less than 1 on this lattice, there will be closed lines (where the value of the line itself is more than the threshold). With a threshold more than 2, there will be open lines (where the value of the line is less than the difference between the threshold and 2), but it is unclear what happens with thresholds between 1 and 2. The critical threshold here is probably 1, 3/2, or 2. If the triangular lattice is like the square lattice in that a little bit of freedom will be enough to percolate part of the time, the critical threshold will be 1. If it is like the planar integer lattice and

needs half its vertices to be open, the critical threshold will be 3/2. If it needs more freedom, the critical threshold will be 2. At this point, the only clear information is that with a threshold t between 1 and 2, with probability 1 there will be open and closed vertices on every line.

7.2 Distribution Generalizations

Another way to generalize is by changing the distribution. So far, the value $a_i(j)$ is considered uniform on $[0, 1]$. What if it takes on another distribution?

Recall in Section 1.3 the motivation for looking at coordinate percolation. We have n programs, each with an infinite list of upgrades, and each upgrade uses a random amount of system resources. We've been assuming that the amount of system resources needed by any upgrade is given by a uniform random variable on $[0, 1]$, representing a percentage of the resources of the average system. However, that an upgrade would change the required percent of resources from 5% to 73% is difficult to imagine. One way to fix this is to adapt the model. We'd like to be able to say that program number i will use resources based on some distribution ρ_i, or even that all the programs will use resourced based on only one distribution ρ. This is beyond what we can accomplish at this time. However, for one collection of related distributions we can give an answer.

Theorem 7.2.1. *Suppose $\alpha_i \colon \mathbb{N} \to [l_i, l_i+d]$ for each $1 \leq i \leq n$. Define a function c_τ where $c_\tau(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathcal{F}_n$ is the configuration where a vertex $v = (v_1, v_2, \ldots, v_n)$ is open if and only if $\alpha_1(v_1) + \alpha_2(v_2) + \cdots + \alpha_n(v_n) < \tau$. Then the probability $\phi(\tau)$*

that a uniformly random $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ maps by c_τ to a configuration which percolates is given by

$$\phi(\tau) = \theta\left(\frac{\tau - \tau_0}{d}\right)$$

where $\tau_0 = \sum_{i+1}^{n} l_i$.

Proof. If $\alpha_i(j)$ is uniform in $[l_i, l_i + d]$ then $a_i(j) = (\alpha_i(j) - l_i)/d$ is uniform in $[0, 1]$ and $\alpha_1(v_1) + \alpha_2(v_2) + \cdots + \alpha_n(v_n) < \tau$ iff $a_1(v_1) + a_2(v_2) + \cdots + a_n(v_n) < (\tau - \tau_0)/d$. Therefore, $c_\tau(\alpha)$ is the same configuration as $f_t(a)$ where $t = (\tau - \tau_0)/d$ and $a = (a_1, a_2, \ldots, a_n)$. So the probability of percolation for a uniform α is the same as the probability $\theta(t)$ of percolation for a uniform a. □

This only supplies the probability of percolation for uniform distributions, and only when all the directions are allowed to take on values with ranges that are the same time. The most general probability of percolation would be the probability that, given a sequence of functions $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, where $\alpha_i(j)$ is a random variable with distribution function ρ_i, the configuration $\sigma \in \mathcal{F}$ percolates, where a vertex $v = (v_1, v_2, \ldots, v_n)$ is open in σ iff $\alpha_1(v_1) + \alpha_2(v_2) + \cdots + \alpha_n(v_n) < t$. To answer this question, we would like to use that fact that a random variable X with probability distribution $\rho\colon \mathbb{R} \to \mathbb{R}$ is equivalent to the random variable Y with uniform distribution in $[0, 1]$ given by $Y = \int_{-\infty}^{X} \rho(y)dy$. This translates all distributions to the uniform distribution in $[0, 1]$, but doesn't help because the condition on open points, namely that $\alpha_1(v_1) + \alpha_2(v_2) + \cdots + \alpha_n(v_n) < t$, does not translate.

However, if we have a distribution with only one infinite tail, for example the

exponential distribution, the probability of percolation may be more tractable. In coordinate percolation with an exponential distribution on the random variables $a_i(j)$, the probability of percolation is 0. This is because for each i there is a j_i so that $a_i(j_i) > t$, where t is the given threshold. The vertices with j_i as their ith coordinate are thus closed, forming a closed box containing the origin. This will hold for any distribution with an infinite tail toward $+\infty$ and not toward $-\infty$. Distributions with infinite tails toward $-\infty$ and not toward $+\infty$ have not been explored.

7.3 Operation Generalizations

What happens if we allow other operations than addition? For example, given $(a, b) \in \Omega_2$, we could define a vertex (i, j) to be open if $a(i) \cdot b(j) < t$. This is another generalization, but can be rephrased as a distribution generalization. For this example, the condition $a(i) \cdot b(j) < t$ is equivalent to the condition $\log a(i) + \log b(j) < \log t$ and, since a distribution on $a(i)$ induces a distribution on $\log a(i)$, if we know the probability of percolation with strange distributions we will know the probability of percolation with this operation. Therefore, it is reasonable to look for the probability of percolation with generalized distributions before looking at the probability of percolation with generalized operations.

Notation List

$\mathbb{N}^n$	Nonnegative portion of the n-dimensional integer lattice.
$l(\varphi)$	The length of a path φ.
$\varphi(i)$	The ith vertex in the path φ. Indexing of a path starts at 0.
$\varphi_j(i)$	The jth coordinate of the ith vertex of the path φ.
$[n]$	The set $\{0, 1, 2, \ldots, n-1\}$.
Ω_n	The sample space for coordinate percolation. $\{(a_1, a_2, \ldots, a_n) \mid a_i \in \mathbb{N} \to [0,1]\}$ with uniform distribution.
$\mathrm{val}_\sigma(v)$	$a_1(v_1) + a_2(v_2) + \cdots + a_n(v_n)$ where $v = (v_1, v_2, \ldots, v_n) \in E(\mathbb{N}^n)$ and $\sigma = (a_1, a_2, \ldots, a_n) \in \Omega_n$.
$\mathcal{F}_n$	The configuration space $\{\sigma \colon E(\mathbb{N}^n) \to \{0,1\}\}$.
f_t	A function $\Omega_n \to \mathcal{F}_n$ defined by $f_t(\sigma)(v) = 1 \Leftrightarrow \mathrm{val}_\sigma(v) < 1 + t$.
$\theta(t)$	A probability function.
P	Sample space for a sample problem (Section 3.1).
$P(t)$	Configuration space for a sample problem (Section 3.1).
ω_t	Function $P \to P(t)$.
Ω_t	Percolation function $\Omega_n \to \{0,1\}$.
ρ_t	Probability function on the configuration space $\mathcal{F}_n$.

$\Omega_t(m,n)$	A set of ordered pairs of functions: a reduced sample space.
A	$\{(m,n,\sigma)\mid m,n \in \mathbb{N}, \sigma \in \Omega_t(m,n)\}$: a reduced sample space.
f	Percolation preserving function $\Omega_2 \to A$.
p_A	The probability distribution for the reduced sample space A.
$\hat{f}$	The induced percolation map on A.
$P(m,n)$	Percolation probability in a specific case.
ρ	Involution of the reduced sample space A.
g_t	Percolation preserving function $\Omega_t(m,n) \to S_{m+n}$.
S_n	Group of permutations on n elements.
$\mathrm{rev}(a)$	The reverse of a word a.
φ^T	The transpose of a schedule φ.
$\preceq$	Worm-order relation symbol.
$a +^{\varphi} b$	The sum word associated with a given schedule φ.
$a \vee^{\varphi} b$	The join word associated with a given schedule φ.
$w(n)$	The number of worms on the alphabet $[n]$.
M_n	The transition matrix at time n.
$\lambda_n, \overline{\lambda_n}$	Eigenvalues not equal to 1 of M_n.
$\Lambda_n, \overline{\Lambda_n}$	Product of the eigenvalues of $M_2, M_3, \ldots, M_n$.
v_n, w_n	Eigenvectors corresponding to eigenvalues $\lambda_n, \overline{\lambda_n}$, respectively, of M_n.
Q_n	Probability that we are not in the absorbing state after time n.

Index

analytic, 55

backtracking, 27, 29

bond percolation, 2

  independent, 2

bone, 30

closed box, 32, 34

closed vertex, 11, 13

cluster size, 6, 17

  expected, 17

configuration space, 13, 24, 52

continuous, 44, 57

coordinate percolation, 3, 8, 11, 13, 19, 24

  collision-based, 3

  comparison-based, 4

  oriented, 30

critical exponent, 7, 17

critical phenomenon, 6

critical probability, 6, 19

critical threshold, 19, 37

derivative, 57

distribution, 110

does not percolate, 14

exponential distribution, 112

forbidden configuration, 19, 23, 30

fulcrum, 75

geometric distribution, 50, 93

independent percolation, 19

integer lattice, 12, 104

  directed, 17

integral equation, 44

join, 30, 81

maximal chain, 30
meet, 30, 81
modular lattice, 30

non-negative correlation, 88

open cluster, 13, 18
    infinite, 13
open line, 18, 32, 39, 40
open path, 13
open vertex, 11, 13
operation, 112
oriented percolation, 18

path, 12
    length of, 12
percolates, 14
percolation probability, 14, 52, 93, 99
    integral equation, 42
permutation group, 53
phase transition, 6

sample space, 13, 24
schedule, 66
site percolation, 2
    independent, 2, 11
sum worm, 76
symmetric group, 87, 95

threshold, 9, 13
transpose schedule, 66
triangular lattice, 108

witness, 67
worm, 71
worm order, 53, 66, 84
worms, number of, 84

Bibliography

[1] P. N. Balister, B. Bollobás, & A. M. Stacey, Dependent percolation in two dimensions. *Probab. Theory Related Fields* **117** (2000), no. 4, 495–513.

[2] G. R. Brightwell, & P. Winkler, Submodular Percolation (preprint).

[3] J. W. Brown, & R. V. Churchill, *Complex Variables and Applications* [7th Edition], McGraw-Hill Book Co., New York, 2004.

[4] D. Coppersmith, P. Tetali, & P. Winkler, Collisions among random walks on a graph. *SIAM J. Discrete Math.* **6** (1993), no. 3, 363–374.

[5] P. Diaconis, & D. Freedman, On the statistics of vision: the Julesz conjecture. *J. Math. Psych.* **24** (1981), no. 2, 112–138.

[6] P. Gács, The clairvoyant demon has a hard task. *Combin. Probab. Comput.* **9** (2000), no. 5, 421–424.

[7] P. Gács, Clairvoyant scheduling of random walks. *Proc. of the 34th Annual ACM Symp. on Theory of Comput.,* 99–108 (electronic), ACM, New York, 2002.

[8] P. Gács, Compatible sequences and a slow Winkler percolation. *Combin. Probab. Comput.* **13** (2004), no. 6, 815–856.

[9] G. Grimmett, *Percolation.* Second edition. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 321. Springer-Verlag, Berlin, 1999.

[10] P. Winkler, Dependent percolation and colliding random walks. *Random Structures & Algorithms* **16** (2000), no. 1, 58–84.

Department of Mathematics, Dartmouth College, Hanover, New Hampshire 03755

E-mail address: lizz.moseman@gmail.com

URL: http://www.math.dartmouth.edu/~moseman/