APPROXIMATION ALGORITHMS FOR STOCHASTIC COMBINATORIAL OPTIMIZATION, WITH APPLICATIONS IN SUSTAINABILITY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Gwen Morgan Spencer

May 2012

APPROXIMATION ALGORITHMS FOR STOCHASTIC COMBINATORIAL OPTIMIZATION, WITH APPLICATIONS IN SUSTAINABILITY

Gwen Morgan Spencer, Ph.D.

Cornell University 2012

As ecologists and foresters produce an increasing range of probabilistic data, mathematical techniques that address the fundamental interactions between stochastic events and spatial landscape features have the potential to provide valuable decision support in the sustainable management of natural resources. The heart of this thesis explores two models motivated by pressing environmental issues: limiting the spread of wildfire and invasive species containment. We formulate stochastic spatial models in graphs that capture key tradeoffs, and prove a number of original optimization results. Since even deterministic cases in highly-restricted graph classes are NP-Hard (that is, they can not efficiently be solved to optimality), our studies focus on approximation algorithms that efficiently produce solutions which are provably near-optimal.

Our models also represent natural generalizations of ideas in the optimization and computer science literature. In particular, while much recent attention has been devoted to questions about connecting stochastically chosen sets, our applications in sustainable planning suggest extensions of deterministic graph-cutting models; we explore novel problems in *stochastic disconnection*.

BIOGRAPHICAL SKETCH

Gwen Morgan Spencer was born on May 2, 1983 in Redmond, Washington. Gwen attended public schools in Washington State for elementary, middle, and high school, graduating from Mount Rainier High School in Des Moines, WA in 2001. From 4th through 12th grade Gwen's love of mathematics grew through involvement in her schools' math teams which were generously organized by volunteer parents and teachers. At Harvey Mudd College (Claremont, CA), Gwen chose a major in Mathematics, graduating with high distinction in 2005. Next, Gwen spent a year independently traveling the world interviewing women scientists as a Watson Fellow. Returning from this great adventure, Gwen came to Cornell University to learn about something called "Operations Research." As it turns out, this new thing involved plenty of mathematics, and a beautiful new set of ideas about algorithms. Six years after her arrival, in May 2012, Gwen expects to receive her Ph.D. in Operations Research from Cornell University, following the M.S. in Operations Research she earned in May 2010.

First, this thesis is dedicated to my family: my parents, Guy and Gretchen, and my younger sister, Gillian. Secondly, it is dedicated to the great teachers and thinkers I have had the honor to learn from and with.

ACKNOWLEDGEMENTS

First, I want to thank my advisor, David Shmoys. David has been incredibly generous with his time and advice over the years, and empirically speaking, every optimization problem becomes more interesting when he presents it. David has taught me a great deal: in terms of mathematical content, and even more significantly, about the process of doing and working in academic research. It has been a real pleasure.

I also owe thanks to the members of my committee: Robert Kleinberg and David Williamson. Both Bobby and David taught wonderful courses that formed the foundation of my interest in algorithmic topics and have been very encouraging to me personally. I learned a tremendous amount about teaching by taking David's excellent classes, and observing Bobby's insightful questions in the Theory Seminar has made me a better question asker.

My classmates in Operations Research have been a constant source of support. I am so thankful for the encouragement and candid perspectives of the students ahead of me, the camaraderie and community of students in my year as we adventured through the core courses, and the friendship and enthusiasm of the fresh PhD students as we older students started to feel a bit lethargic. In particular, the support of Kathleen King, Tim Carnes, Collin Chan, and Gema Plaza-Martinez has been very dear to me.

I want to thank my friend and coauthor, Igor Gorodezky, for his encouragement, mutual mathematical enthusiasm, and his wonderful sense of humor about some of the stranger features of graduate-school life. An essential component in surviving the last few years have been the friendships I've made while living at the Gamma Alpha Graduate Scientific Society. I am particularly grate-

ful to know Sarah Iams. More broadly, the friendly Theory and CAM communities at Cornell have been very supportive and intellectually stimulating.

I'd also like to thank my undergraduate research advisor, Francis Su, for starting me on this wonderful research path.

Finally, the NSF has generously supported my work in many ways: through a Graduate Research Fellowship, grants no. CCR-0635121, DMS-0732196, CCF-0832782, CCF-1017688, and their broader support for the Institute for Computational Sustainability which has been an invaluable resource for me at Cornell.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Decision makers in natural resources face complex spatial planning problems which fall outside the realm of classical results in Operations Research. While foresters and ecologists model many processes in graphs, classical optimization models in graphs often have inputs that are known in a deterministic way (or have at most one stochastic feature) and assume that actions taken by the planner have exactly the intended effect. In contrast, the scientific understanding of ecological and environmental systems almost inevitably contains several elements of uncertainty: scientific data rarely provide deterministic inputs, the effect of actions taken by the planner have estimated (not deterministic) consequences, and the amount of information about the system changes over time. In many settings related to ecological processes the interplay between spatial and stochastic features is fundamental: it cannot be abstracted away without seriously compromising the relevance of the study. For these reasons, planning problems in the sustainable management of ecological systems are a natural source of novel and aesthetically-interesting stochastic optimization problems in graphs. In addition to expanding the boundary of stochastic optimization, the development of mathematical tools and understanding of these stochastic spatial problems has the potential to significantly impact actual practice.

This Introductory Chapter covers the technical chapters in the order they appear. Chapter 2 describes a graph model for limiting the spread of wildfire through a mix of preventive and realtime action. Chapter 3 focuses on a model for invasive species containment which expands on several ideas from Chapter 2. In both cases, we mention computational hardness and our positive algorithmic approximation results. Last, Chapter 4 describes an earlier project on the relationship of stochastic and adversarial variants of the Traveling Salesman Problem that motivated some of the qualitative questions which shaped the studies in Chapters 2 and 3.

A dramatic feature of managing forests and natural areas is limiting the spread of wildfire. Traditionally, the vast majority of wildfire suppression is conducted once the fire has broken out; the fire is fought aggressively to limit the duration and spatial extent of the event. A natural consequence of this aggressive-suppression policy is that flammable material builds up on the landscape: dead trees and dry brush (which might usually be burned off in regular cycles) dangerously collects in forests and wildlands, creating the potential for fires that spread further and faster. The recent dramatic increase in large and catastrophic wildfires and ballooning Forest-Service expenditures for wildfire-fighting ($+350\%$ over the last two decades [20]) have motivated a conversation among foresters (in academic and government agencies) about whether more sustainable landscape management is possible. In particular, policy initiatives proposed consideration of preventative management strategies like small-scale preventive burns and mechanical underbrush thinning that remove excess flammable material in advance of the wildfire season. Two approaches aiming to optimize how limited budgets for such measures could be deployed were published in the forestry literature [18],[49].

Exploring the placement of preventative fuel reductions seemed mathematicallypromising for two key reasons. First, the forestry literature showed good consensus around models for wildfire spread, and these models had already been translated from the continuous world to good (and efficient) approximations on graphical grids. In particular, this modeling approach was widely distributed in the US Forest Service through two key decision-support softwares developed and supported by the Rocky Mountain Forest Research Station: Farsite and Flammap [17]. Second, foresters were producing probability distributions on wildfire ignition sites. These distributions were used as input to simulate burn probabilities in Flammap and also as input to simplified Integer-Programming models that aimed to optimize the location of preventative fuel treatments [49].

Considering the existing work on preventive fuel removal, we identified two key challenges. First: take advantage of the valuable probabilistic predictions about ignition site while simultaneously respecting the spatial continuity of fire spread through discrete graphical landscapes. Second: create and analyze a model that captures the appropriate balance and coordination between preventative fuel removals and traditional fire-fighting capabilities that can be implemented once the fire location is known (in both [18] and [49] preventive action is planned as if no realtime fire-fighting is available to supplement it).

To address these goals we introduced a new family of stochastic optimization problems in graphs [44]. The most general version is as follows. Given a graph in which each node has some value, each edge has some cost, and the input includes a distribution over ignition nodes in the graph, spend a specified budget removing edges from the graph (either before the realization of the ignition node, or afterwards at increased cost), so that the total value of nodes not-reachable from the ignition node is maximized. Since foresters care about making decisions on large and varied landscapes, the computer-science theory focus on methods whose running times scale polynomially with the instance size was a natural match for this wildfire application.

A thorough search of the optimization and computer science literature revealed substantial recent interest in the deterministic and single-stage special cases of our model [27], [15]: protecting a network from a threat by thinning the network is an attractive concept for many interdiction-type applications (for example, isolating infections that spread according to social contacts or viruses in a network of computers). This previous work showed that even these base cases in simple graph classes are weakly NP-Hard (reducing from the knapsack problem) [27].

Since this hardness result implies that efficient algorithms to solve the problem exactly do not exist (assuming P $\neq$ NP), we focused our study on *approximation algorithms*: efficient algorithms with performance provably-close to the performance of the optimal solution. Formally, letting OPT denote the value of an optimal solution:

Definition 1.0.1. *An algorithm is an α-**approximation** if, for every input instance, the algorithm produces in polynomial time a feasible solution of value at least αOPT.*

There is also a useful notion of an algorithm which produces solutions that are both near-optimal and near-budget balanced:

Definition 1.0.2. *Given a budget B, an algorithm is an (α, β)-**approximation** if, for every instance, the algorithm produces in polynomial time a feasible solution of value at* *least αOPT, while spending at most βB.*

Finally, there is a class of approximation algorithms solve arbitrarily accurately:

Definition 1.0.3. *An algorithm is a* ***Polynomial-time Approximation Scheme (PTAS)*** *if, for any constant $\epsilon > 0$ the following condition holds: For every input instance, the algorithm produces a feasible solution of value at least $(1-\epsilon)OPT$ in time which is polynomial in the instance size, and also polynomial in $\frac{1}{\epsilon}$.*

In the simplest base cases, arbitrarily-accurate approximations are possible. The foundation of these methods are psuedopolynomial-time exact dynamic-programming algorithms that are converted to efficient schemes by rounding the input (and bounding the resulting loss in performance). Via a simple reduction we extend the existing PTAS of [27] in graphs of bounded treewidth for the deterministic ignition-point case to a PTAS in graphs of bounded treewidth for the deterministic ignition-set case. Applying some careful partial-enumeration, we extend to demonstrate a PTAS in trees for the stochastic case in which the number of scenarios is constant.

As we generalize to more complex cases, we employ an increasingly sophisticated set of approximation techniques to obtain a hierarchy of approximation results. For the limiting stochastic case in which no realtime action is possible, we give a $(1-(1-1/\delta)^{\delta})$-approximation algorithm in trees for the case of probabilistic ignition from a single source (δ is the diameter of the tree, the asymptotic limit of this approximation guarantee is $(1-\frac{1}{e})$, or approximately 63%). This result is proved by reducing to the Maximum Coverage subject to a Knapsack Constraint Problem of Ageev and Sviridenko [1]. For the 2-stage stochastic model in which actions may either be taken in advance of the ignition based on probabilistic information, or after the single ignition point is known at inflated cost, we give a $(1 - (1 - 1/2\delta)^{2\delta})$-approximation in trees which violates the budget by a factor of at most 2. Notably, the inflation in the second stage can vary across scenarios and edges. Our approach here is to solve a natural linear-programming relaxation with a more complex feasible region than that considered by Ageev $\&$ Sviridenko [1]; we are able to extend their pipage-rounding analysis to reduce the number of fractional variables: this requires additional specifications about which pairs of fractional decision variables may be rounded against each other and a careful treatment of the larger number of fractional variables that remain at the end of the pipage stage. We explain how the flexible form of our LP can be used to capture more spreading-process-specific features through a careful tuning of the input. We also give a 0.387-approximation which is budget-balanced for the 2-stage stochastic model, and some results for a k-stage extension.

In some cases we can extend to general graphs with an additional $O(\log n)$ loss in budget-balancedness. All extensions from trees to general graphs employ the probabilistic capacity-preserving mapping of Räcke in the standard way (see [15]): approximate the costs of all cuts in the graph by a distribution over trees, solve a suitably modified instance in each tree, translate solutions back to the original graph, select the best solution.

For an extension in which fire transmission across edges is probabilistic and depends on the level of investment in removing the edge (assuming independence of edge realizations), we give a $(1-1/e)$-approximation algorithm in trees by proving submodularity of the objective function.

Our multistage and probabilistic-transmission results in trees also hold for analogous generalizations of the Maximum Coverage with Knapsack Constraint problem (MCKP) in which elements may *fail* independently with probability that depends on the level at which we invest in them, and the objective is to maximize the expected weight of the sets which nontrivially intersect the realized elements. For probabilistic element-failure MCKP, our guarantee matches the asymptotic guarantee for the deterministic element case from Ageev & Sviridenko [1].

In addition to our positive algorithmic results, our approach yielded qualitative insights. For example: compared to methods that try to solve the preventative and realtime stages independently, the IP-based solution quality begins to dominate strongly as the variance of the ratio of realtime costs to preventive costs increases. The qualitative lesson for planners is that coordination between decisions about prevention and real-time measures becomes increasingly vital when some options may effectively expire by becoming prohibitively expensive (or impossible) in realtime. No previous literature on preventative fuel removal had attempted to coordinate preventative actions and realtime decisions; our work identifies important features that determine how and when this coordination is important, and points out that the optimal balance between these investments is specific to the problem instance (and cannot be determined exogenously).

Another theme that emerges from Chapter 2 echoes results concerned with connecting sets that are chosen stochastically: stochastic fragmentation is much simpler in trees. In particular, our results always make critical use of tree structure, and our results in general graphs all rely on a general method of extension from trees to graphs which requires only that connectedness in the tree translates directly to connectedness in the graph (this fails to hold for the probabilistic edge-removal case). From a theoretical perspective, though our results in general graphs resort to logarithmic budget approximation, our strong results in trees can be seen as partial progress towards more general settings: understanding methods in trees is often a first step towards understanding results in more general graph classes.

In fact, results in trees are also directly relevant to the management of several ecological systems: trees are used to study invasive-species spread through stream and river systems. For example, Graeme Cumming's Nature-Conservancy-funded report ″Habitat Shape, Species Invasions, and Reserve Design: Insights from simple models″ compares the course of simulated invasions in branching stream systems [14]. Invasions of tree-landscape topologies also occur in the riparian-vegetation zones bordering river systems. For example, Tamarisk is an exotic noxious weed that invades such zones, driving out local wetland plants; the Colorado Department of Agriculture has suppressed Tamarisk invasion using the tamarisk leaf beetle, *Diorhabda elongata* along the Dolores, Colorado, Yampa and Green Rivers) [13].

Having observed the relative ease of analyzing stochastic fragmentation in trees (compared with general graphs), and the connection to invasive species containment, it was natural to consider whether there were additional features of planning problems associated with invasive species containment that could be modeled while retaining theoretical traction.

In speaking with Robert Haight (a Research Forester with the US Forest Service) about his work [32], [33] towards a plan for the containment of the Emerald Ash Borer (an invasive beetle that is deadly to all Ash tree species) in the Minneapolis, St. Paul area, several interesting issues emerged. Foresters are interested in models with much more explicit timestep evolution; this captures more information about the expanding shape of the process and allows more detailed investigation of the advantages of concentrating action and resources early. Funding and resources to fight invasions may become available gradually as societal awareness and political consensus form around the advantages of suppression. A soft constraint on resource availability is that counties may be averse to spending towards containment efforts which are not within their boundaries, preferring to reserve these resources for containment efforts once the invasive species has reached their jurisdiction (a main message of the computational study in [33] is that this strategy is dangerously short-cited: simulation results point to huge increases in damages for currently-unaffected counties if initial containment efforts are underfunded). Finally, invasive species containment is an inexact (though sometimes probabilistic) science in many of its features. There are many documented cases of ”jump events” in which an invasive species spreads across a barrier intended for containment (sometimes a zone in which the known host habitat of the invasive species has been removed, sometimes a water feature, etc). Ecologists have started to model these types of jump events probabilistically [35]. Another stochastic feature explored in the ecological literature is that introductions of control species (intended to inhibit the spread of the invasive species) often fail to establish viable colonies [43]. The probability of establishment can sometimes be estimated as a function of the size of the seed colony [43] and may also depend on stochastic weather conditions [38] and suitability of local habitat for the control species.

In Chapter 3, we begin the theoretical exploration of this second source of stochasticity (colony-establishment failure) by generalizing Hartnell's Firefighter Problem [26] which concerns the containment of an infection outbreak in a graph when the amount of action per time step is limited. During each time step in the Firefighter Problem all neighbors of an infected node become infected. The planner has the option to vaccinate k nodes per time step (so that they will never become infected). In our generalization, vaccination fails with probability that can depend on both time and location of deployment, and the resources available in each timestep need not be uniform. Our generalization gives a very general model that describes some of the key features of containing the spread of an invasive species through the deployment of biological control species whose colony-establishment succeeds probabilistically. We address cases in which the control species also spreads through the landscape at the same rate as the invasive species, and in which the control species is effective only at the point of colony establishment. Our spreading case generalizes a version of the spreading Firefighter Problem introduced in the social-networking literature to describe rumor-spreading (and rumor-combatting) in social contact networks [3].

We give reductions of the spreading case in general graphs and the non-spreading case in trees to a Maximum-Coverage with-probabilistic-element-failure objective (defined in Chapter 2) subject to a partition-matroid constraint. We exploit the Maximum Coverage property (which is strictly stronger than submodularity) to two key ends. First, to observe a deterministic (1-1/e)-approximation for the base case where vaccination (a.k.a. colony establishment) is perfectly effective and spreading (matching the previous-best randomized approximation guarantee for that problem from [3]). Secondly, in the probabilistic-failure case, we can evaluate (in polynomial-time) the objective function value (which is not necessarily possible if only submodularity holds), such that an existing algorithm of Calinescu, Chekuri, Pál and Vondrak [11] gives a randomized (1-1/e)-approximation. Our reductions recognize that the partition-matroid constraint need not be wasted constraining action per time step (this is captured by making failure-probability a function of time); instead it is used to greatly expand the descriptive power of the model without losing (almost any) theoretical traction.

Chapter 3 builds on several key questions of Chapter 2. First: Can probabilistic predictions of ecologists be harnessed by optimization methods towards planning problems? Second: What properties of the objective make these problems easier in trees, and can these properties be forced in wider classes of graphs, or through additional model features (e.g. introducing spreading)? Both Chapters focus on the idea of stochastic disconnection, and look at models where this disconnection task is accomplished over stages that emphasize trade-offs between acting early (with greater uncertainty: either in the lack of knowledge about ignition site or in the low likelihood that attempted action will be successful) and acting later with greater certainty (but once action has become more expensive, or the spatial extent of the invasion has grown). Essentially, the algorithms must identify edge removals (or node vaccinations) that contribute in a valuable way towards cuts which are useful according to different realizations of randomness (in ignition point, or in other implemented vaccinations). Implicitly, the algorithm produces some valuation of edge removals/nodes vaccination according to cutting power across many scenarios (weighted according to distribution). In some sense, at every decision point the algorithm picks a solution that is *versatile* across the coming possible realizations (in an average case sense).

Many of the stochastic-disconnection investigated in Chapters 2 and 3 grew out of earlier joint work included in Chapter 4 [22]. In particular, though the *a priori* Traveling Salesman Problem (which seeks a master ordering that performs well on average when shortcut to a subset scenario realized from a distribution) appears to be about a tour of a stochastically-chosen set, the heart of the best positive result [45] is an argument about stochastic connection (where ordering is not important). Specifically, the constant approximation of Shmoys and Talwar depends on a rent-or-buy idea: when membership of nodes in the scenario set is realized independently, a set of "bought" connections valuable across scenarios (and not too costly) can be accessed by sampling from the distribution. In this restricted setting (independent occurrence of nodes), a random sample effectively identifies a versatile and affordable "backbone" for stochastic connection; this backbone is subsequently augmented to connect the subset realized from the distribution. This sampling approach cannot be extended to general distributions; the reason lurks very fundamentally in the adversarial case of the Traveling Salesman Problem (named the "Universal Traveling Salesman").

In Chapter 4 we show that there exist metrics where under this stronger notion of adversarial versatility (that is, where an ordering is only considered as good as its performance on the subset where it performs the worst), every possible tour has performance $\Omega(\log n)$. In particular, given any fixed ordering of the nodes of the shortest path metric in a Ramanajun graph, we use a probabilistic argument to prove the existence of a node subset with optimal tour logarithmically shorter than the performance of the fixed ordering. Our lower bound for the universal TSP improves on the previous best bound of Hajiaghayi, Kleinberg, and Leighton, who showed that the competitive ratio of the $n \times n$ grid is $\Omega(\sqrt[6]{\log n / \log \log n})$.

Next we show that for a large class of combinatorial optimization problems that includes TSP, a bound for the universal problem implies a matching bound on the approximation ratio achievable by deterministic algorithms for the corresponding black-box *a priori* problem (where no assumptions are made about the form of the distribution and it is observed only through sampling). This theorem follows from an information-theoretic argument and provides an explicit testimonial to the power of randomization: it implies that the approximation ratios guaranteed by several existing randomized algorithms for *a priori* problems cannot be matched by deterministic algorithms. As a consequence, our lower bound of $\Omega(\log n)$ for the universal TSP implies a matching lower bound for the black-box *a priori* TSP. This shows that independence is critical to Shmoys & Talwar's $O(1)$-approximation in the case of independent activation probabilities: no algorithm for the general problem that acts deterministically on the black-box samples can achieve $o(\log n)$-approximation.

The work in Chapter 4 raised a number of qualitative questions. When do there exist partial solutions to stochastic optimization problems that are both versatile (or cheaply updatable) and affordable? For example, what restrictions on the space of distributions, the objective form, and the class of graphs are needed to ensure such existence? These questions are relevant for problems that are explicitly multistage, but also for problems which are single-stage, but where such a property can be used analytically to bound constructed-solution cost. Next, when are such partial solutions efficiently-computable, or when can provably-good approximations to such partial solutions be computed efficiently? In proving guarantees, when is the the algorithm's ability to randomize critical and when can good results be obtained via deterministic methods? Finally, while several works gave approaches to these questions for stochastic connection (for TSP and other problems): could similar questions be asked and answered when the goal was not connection, but disconnection? These ideas shaped the studies described in Chapters 2 and 3.

Each of Chapters 2-4 includes a topic-specific Summary and Introduction sections that expand substantially on what is covered here in Chapter 1. The necessary technical definitions to discuss existing prior work and the full range of our results in depth appear there. Also, the chapters are largely self-contained: for the most part, they need not be read in sequence.

CHAPTER 2

APPROXIMATION ALGORITHMS FOR FRAGMENTING A GRAPH AGAINST A STOCHASTICALLY-LOCATED THREAT

The work in this chapter is joint with David Shmoys and appeared in a preliminary form in the Proceedings of the Workshop on Approximation and Online Algorithms, 2011 [44].

2.1 Summary

Motivated by issues in allocating limited preventative resources to protect a landscape against the spread of a wildfire from a stochastic ignition point, we give approximation algorithms for a new family of stochastic optimization problems. We study several models in which we are given a graph with edge costs and node values, a budget, and a probabilistic distribution over ignition nodes: the goal is to find a budget-limited set of edges whose removal protects the largest expected value from being reachable from a stochastic ignition node. In particular, 2-stage stochastic models capture the tradeoffs between preventative treatment and real-time response. The resulting stochastic cut problems are interesting in their own right, and capture a number of related interdiction problems, both in the domain of computational sustainability, and beyond.

In trees, even the deterministic problem is (weakly) NP hard: we give a PTAS for the stochastic single-stage case in trees when the number of scenar-

ios is constant. For the 2-stage stochastic model in trees we give a $(1 - \frac{1}{e}) < (1 - (1 - 1/2\delta)^{2\delta})$-approximation in trees which violates the budget by a factor of at most 2 (δ is the tree diameter), and a 0.387-approximation that is budget-balanced. For the single-stage stochastic case in trees we can save $(1-(1-1/\delta)^{\delta})OPT$ without violating the budget. Single-stage results extend to general graphs with an additional $O(\log n)$ loss in budget-balancedness. Multistage results have a similar extension when the number of scenarios is constant. In an extension of the single-stage model where both ignition and spread are stochastic we give a $(1 - \frac{1}{e})$-approximation in trees. Our approximation guarantees in trees also hold for multistage and probabilistic-element-failure generalizations of the Maximum Coverage with Knapsack Constraint problem (MCKP).

2.2 Introduction

Increasing frequency of catastrophically-damaging wildfire events has stimulated interest among foresters and land managers in effective use of preventative fuel reductions. Traditional fire suppression policy has focused almost exclusively on realtime firefighting (once the fire has broken out), but preventative fuel reductions such as dead-brush removal, small-scale controlled burns, and crown raising can be applied in advance to slow or stop the spread of wildfires. Recent wildfire modeling literature has used historical and scientific information to estimate a distribution of wildfire occurrence in which both the ignition site and the wind direction can vary [18],[49].

The planning problem of how to allocate limited resources across preventative and realtime stages, and where to distribute preventative resources using probabilistic information motivates a natural new family of budgeted stochastic optimization problems that fragment (or cut) a landscape graph to isolate a stochastically occurring ignition point. A key feature is the tradeoff between spending preventively when only distributional knowledge is available and spending at increased cost once a fire has broken out. We explore a number of model variants motivated by these circumstances. Studying this family of problems through the lens of efficient approximation, we give constant bicriteria approximations in trees, and a budget-balanced constant approximation for the limiting case in which real-time actions become prohibitively expensive. Our techniques also yield new approximation results for multistage stochastic extensions of the budgeted Maximum Coverage problem. The theme of our models (protecting a network from the spread of a stochastic outbreak of a harmful diffusive process) has other important environmental applications (e.g., containing invasive species over land or through water systems).

Results. In trees, the problem is (weakly) NP hard even when there is a single ignition point that is known deterministically [27] (the Knapsack Problem is a special case). An existing PTAS in graphs of bounded treewidth for the deterministic ignition-point case extends immediately to a PTAS in graphs of bounded treewidth for the deterministic ignition-set case. Applying some careful partial-enumeration then allows a PTAS in trees for the stochastic case in which the number of scenarios is constant.

For the 2-stage stochastic model in which actions may either be taken in advance of the ignition based on probabilistic information, or after the single ignition point is known at inflated cost, we give a $(1-(1-1/2\delta)^{2\delta})$-approximation in trees which violates the budget by a factor of at most 2 (δ is the tree diameter). Notably, the inflation in the second stage can vary across scenarios and edges. For the limiting stochastic case in which no realtime action is possible, we give a $(1-(1-1/\delta)^{\delta})$-approximation algorithm in trees for the case of probabilistic ignition from a single source. We also give a 0.387-approximation which is budget-balanced for the 2-stage stochastic model, and some results for a k-stage extension. In some cases we can extend to general graphs with an additional $O(\log n)$ loss in budget-balancedness via the probabilistic cut-capacity approximation result of Räcke [40] as in Engelberg, et al. [15].

For an extension in which transmission on edges is probabilistic and depends on the level of investment in removing the edge (assuming independence of edge realizations), we give a $(1-1/e)$-approximation algorithm in trees.

Our multistage and probabilistic-transmission results in trees also hold for analogous generalizations of the Maximum Coverage with Knapsack Constraint problem (MCKP) in which elements may *fail* independently with probability that depends on the level at which we invest in them, and the objective is to maximize the expected weight of the sets covered by the realized elements. For probabilistic element-failure MCKP, our guarantee matches the asymptotic guarantee for the deterministic element case from Ageev $\&$ Sviridenko [1].

Related literature. The placement of preventative fuel treatments has been addressed in the recent forestry literature. Finney [18] prioritizes spatial fire spread dynamics, limits probabilistic model components, and aims to reduce the rate of spread of the head of fire. Wei et al. [49] considers the objective of reducing expected value lost across a grid-cell landscape by reducing burn probabilities (probabilities computed through simulation); however their IP-based approach is based on a questionable linearity assumption. These approaches produce divergent solution forms, and it is clear that the development of additional mathematical tools and techniques that simultaneously address stochastic and spatial aspects would be useful to decision-makers faced with this important planning problem.

The problems we study have ties to the existing computer science literature. The special case in which the ignition point is known deterministically and there is a single decision stage has been studied as the Minimum-Size Bounded-Capacity Cut problem by Hayrapetyan et al. [27]. They show that the problem is weakly NP-hard in trees by reduction from the Knapsack problem. In general graphs they give two different $(\frac{1}{1-\lambda}, \frac{1}{\lambda})$ bicriteria-approximations for the (expected value burned, budget), and they give a PTAS in graphs of bounded tree width. Engelberg, et al. [15] study a number of budgeted cut problems in graphs including the weighted Budgeted Separating Multiway Cut Problem (wBSMC), which the single-stage (aka, no realtime action) stochastic version of our problem reduces to. They apply Räcke's probabilistic cut-capacity-preserving approximation to reduce to the case of trees, then observe submodularity in trees, and apply [48] to get a $((1-1/e),\ O(\log n))$ bicriteria result. Our LP-based result for the single-stage stochastic version of our problem in trees generalizes to wBSMC in trees giving a slightly stronger $(1-(1-1/n)^n,\ O(\log n))$ bicriteria result.

Techniques. For the deterministic case, a psuedopolynomial-time exact dynamic programming method is converted to an efficient scheme by rounding the input (as in [27]): our extension to general ignition sets is by demonstrating bounded treewidth of a modified input. For the extension with probabilistic-edge transmission, proving submodularity in tree graphs allows application of Sviridenko's [48] result on budgeted maximization of submodular functions. In the multistage-stochastic case, we solve a natural LP with a more complex feasible region than that considered by Ageev & Sviridenko [1], but we are able to extend their pipage-rounding analysis to reduce the number of fractional variables: this requires additional specifications about which pairs of fractional decision variables may be rounded against each other and a careful treatment of the larger number of fractional variables that remain at the end of the pipage stage. All extensions from trees to general graphs employ the probabilistic capacity-preserving mapping of Räcke in the standard way (see [15]): approximate the costs by a distribution over trees, solve a suitably modified instance in each tree, translate solutions back to the original graph, select the best solution. Our techniques also yield similar results for stochastic multistage and probabilistic item-failure extensions of the constrained Maximum Coverage problem.

The Graph Protection Problem: Summary of Main Results		
	restricted graph classes	general graphs (via Räcke)[40]
2-stage stochastic, single source	 trees: $(1-(1-1/2\delta)^{2\delta}, 2)$ Via pipage rounding. *Alternative:* $(0.387, 1)$	 constant number of scenarios* $\Rightarrow$ $(1-(1-1/2n)^{2n}, O(\log n))$
stochastic, single source with (B_1, B_2)	trees: $(1-(1-1/2\delta)^{2\delta}, 1, 2)$ Via pipage rounding.	constant number of scenarios* $\Rightarrow$ $(1-(1-1/2n)^{2n}, O(\log n), O(\log n))$
k-stage stochastic, single source	 trees, restricted partition hierarchy: $(1-(1-1/k\delta)^{k\delta}, 2+\epsilon)$ Via pipage rounding.	 constant number of scenarios* and restricted partition hierarchy $\Rightarrow$ $(1-(1-1/kn)^{kn}, O(\log n))$
1-stage stochastic, single source with probabilistic edges	 trees: $(1-1/e, 1)$ Due to submodularity.	 open
stochastic, single source	trees: $(1-(1-1/\delta)^{\delta}, 1)$ Reduce to MCKP, apply [1].	$(1-(1-1/n)^{n}, O(\log n))$
stochastic with constant support and constant source size	trees: $(1+\epsilon, 1)$	$(1+\epsilon, O(\log n))$
deterministic with arbitrary source size	bounded tree width: $(1+\epsilon, 1)$	$(1+\epsilon, O(\log n))$
deterministic with single source	bounded tree width: $(1+\epsilon, 1)$ [27]	$(1+\epsilon, O(\log n))$ [27]

$*$ *These results require some conditions on the inflation factors. See Section 2.7.*

2.3 2-Stage Stochastic Graph Protection Problem in Trees

The spread of wild fires can be prevented both through advance fuel treatments and through real-time fire-fighting. Our model captures the tradeoff between using resources in advance vs. waiting until the realization of the ignition point is known but operations are more costly.

The input is a connected tree $T = (V, E)$, a non-negative *value* function $v : V \to \mathbb{Z}$, a non-negative *cost* function $c : E \to \mathbb{Z}$, and a *budget* B. A distribution

Π over *source nodes* i is specified. In the first stage Π is known, and it costs c_e to remove edge e from T, in the second phase a realization from Π is specified (say the source is i), and edge e may be removed from T at cost $M^{ie}c_e$. That is: edges purchased in the second stage, once the source is known, have increased cost by a multiplicative *inflation factor* that may depend both on the scenario realized and on the edge.

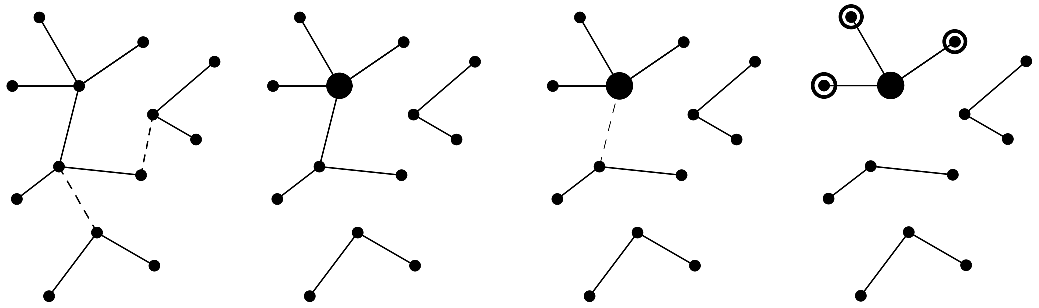

Figure 2.1: In the leftmost graph two edges have been selected to be removed in the first stage (indicated by dashed lines). Next the ignition node is realized: this is shown as the larger black node in the second graph. After the realization of the ignition node additional edges can be selected to be removed during the second stage (third graph). Finally, the fire spreads through the connected component containing the ignition node (4th graph): non-ignition nodes which are lost to fire are indicated by circles.

The total spending on removing edges from T over both phases must be at most B. The objective is to specify a set of edges to buy in the first stage, and then a set of edges to buy in the second stage (depending on the realized source node from Π), such that the expected value not reachable from the realized source node is maximized. We aim to maximize the expected value *protected* from the source. See Figure 2.1. We can contract all edges with costs strictly greater than B since they will not be in any optimal solution.

Special Case 0. Consider the limiting case when all second-stage actions are prohibitively expensive and also Π has support of size 1: this case is the Minimum-Size Bounded-Capacity Cut problem of Hayrapetyan, et al. [27]. They give a PTAS in graphs of bounded tree width and show that this deterministic problem with a single ignition node is NP-hard in trees.

Suppose in this deterministic single-stage case we replace the single ignition point s with a ignition set S. Now the objective is to maximize the expected value protected from every node in S by removing a budget-balanced set of edges.

Theorem 2.3.1. *There exists a PTAS in graphs of bounded tree width for the single-stage deterministic Graph Protection Problem (GPP) with a general ignition set.*

Proof. Consider an arbitrary instance of the single-stage deterministic problem with a general ignition set in graph G of bounded treewidth. To the graph G, add a single super-source s with edges of cost $2B$ to each source in the input. Give the source nodes value 0. Given an arbitrary tree decomposition of G of width θ, a valid tree decomposition for the modified input is just same tree with the node s added to every subset of $V(G)$ corresponding to a node of the tree: this tree decomposition has width $\theta + 1$. So the PTAS of [27] runs in polynomial time on this modified input. Since the PTAS of [27] never buys any of the added edges (they are too costly) the solution it returns on the modified input is a cost-B subset of the original $E(G)$ where any node protected from the super-source in the modified input is protected from every ignition source in the original input, and vice versa. Thus the algorithm: modify the input, run the PTAS from [27] is the PTAS claimed. □

The PTAS asserted in Theorem 2.3.1 for trees can also be produced directly by extending the classic dynamic programming framework for Knapsack. See Section 2.8.1 in the technical addendum.

Applying [27] with an enumeration scheme over a polynomial number of divisions of the tree into source-containing components which can each be modified to act as a single-source deterministic problem, we get:

Theorem 2.3.2. *There exists a PTAS for the stochastic single-stage GPP in trees provided that the size of the support of the distribution* Π *and the size of each ignition set given positive weight by* Π *are bounded by a constant.*

Proof. The algorithm enumerates a large (but polynomial number) of possible solutions, where each solution is constructed in two phases. In the first phase a set of edges are removed from the tree to give an *initial partition* of the tree. In the second phase the remaining budget is spent removing additional edges from the tree under the constraint that no two nodes which both act as sources (under any scenario) and are common to a single partition piece are newly disconnected. The algorithm returns the full solution (among those considered) that protects the most expected value. For any set of initial partitions, this problem is more constrained than the original problem: showing that the optimal solution for the original problem is among the solutions considered for this constrained problem and giving a PTAS for the constrained problem is sufficient to get a PTAS for the original problem.

The algorithm will consider the best solution for each initial partition that corresponds to removing a set of at most $kj - 1$ edges from the tree (where k is the bound on the size of the support of Π and j is the bound on the size of each ignition set). There are at most $(kj - 1)n^{kj-1}$ such partitions (this is a bound on the sum from $i = 1$ to $i = kj - 1$ of n choose i). When k and j are constant, this gives a polynomial number of initial partitions.

For each initial partition the algorithm contracts all source-to-source paths in each partition-piece (to obtain an acyclic graph with a single super-source in each connected component). For each resulting connected component, the algorithm runs the Dynamic Programming fptas for deterministic ignition from the single super-source (with the probability of ignition for a connected component's super-source being the sum of the probabilities of all scenarios which had a source in the corresponding partition piece) with budget B-(the cost of the edges defining the initial partition). This gives a list of undominated pairs for each connected component. By considering every possible combination of undominated pairs in which one undominated pair is selected from the list of each connected component's DP run (where values protected are appropriately weighted according to the probability that the fire breaks out in that connected component), the algorithm obtains the best budget-balanced solution for the fixed initial partition. (Note: since each list has polynomially many entries and there are at most kj such lists the number of combinations is polynomial).

Comparing the best solutions obtained over all initial partitions, the algorithm chooses the best budget-balanced full solution.

The algorithm described produces a budget-balanced solution in polynomial time. For each fixed initial partition, the algorithm computes a $(1-\epsilon)$-approximation of the optimal expected value that can be protected by spending the remaining budget under the constraint of not separating nodes which act as sources (as a result of considering all combinations of undominated pair lists where each list is $(1-\epsilon)$-optimal). It is thus sufficient to show that the optimal solution is feasible for the constrained problem for some initial partition considered by the algorithm.

Assume wlog that T is initially connected. Consider the partition in T which is defined by the optimal solution Y for the original problem (Y is a set of edges of cost at most B). It has some collection of connected components. Some connected components may not contain any nodes which act as sources under any scenario. If a component has this property, add edges of T adjacent to the component back into the graph one at a time until the component no longer has this property. Repeat until no source-less components remain. Note that this procedure does not result in any component mergers in which both components merging contain a node that acts as a source. If the cost of the edges added back to the graph by this procedure is b then the cost of the edges that define the partition we now have is at most $B-b$. Call the set of edges which define this partition in the graph X.

Now we have a partition in which each connected component contains at least one node that acts as a source under some scenario. Thus, there are at most kj connected components. Each partition of T that has at most kj connected components can be defined by the removal of at most $kj-1$ edges from T. Thus, one of the initial partitions the algorithm considers will be X.

Now consider the set of edges $Y \setminus X$. No edge in this set is on a source-to-source path in the partition defined by X (since getting from Y to X involved no mergers of 2 source-containing components). Thus, $Y \setminus X$ is feasible for the constrained problem given initial partition X. We conclude that the algorithm is a PTAS. □

Theorems 2.3.1 and 2.3.2 can be extended to bicriteria approximations for general graphs as in Engelberg, et. al [15]: the guarantees on value protected (expected value protected) are identical (though δ may be as much as n), and the budget is violated by a $O(\log n)$-factor (applying Räcke's result [40] on cut-capacity approximation). We explain more details of this extension to general graphs in Section 2.7.

Theorem 2.3.3. *There exists a bicriteria $(1 - (1 - \frac{1}{2\delta})^{2\delta}, 2)$-approximation algorithm for the 2-stage stochastic Graph Protection Problem in trees provided that each scenario has a single ignition node (δ denotes the tree diameter).*

In general graphs, if there are a constant number of scenarios and some conditions on inflation factors hold, Theorem 2.3.3 can be extended to a $(1 - (1 - \frac{1}{2n})^{2n}, O(\log n))$-bicriteria approximation (the multistage extension requires an application of the Markov inequality to ensure $O(\log n)$-capacity distortion for each scenario under the cut-capacity approximation, details in Section 2.7).

The following proof of Theorem 2.3.3 does not require that the node values are uniform across scenarios, but for notational convenience we will ignore this. This flexibility (and creative use of scenario-dependent edge costs) allows the input form to describe spatial properties of certain types of diffusive processes so that fragmenting the graph has more subtle process-specific implications for value protection than is immediately obvious when considering connectivity (details appear in Section 2.5).

Roughly, the key ideas of the proof follow: the optimal fractional solution to a natural LP for 2-stage GPP acts as a starting point for a rounding algorithm. The rounding algorithm (carefully) chooses two fractional variables and rounds the LP solution along a vector that maintains their weighted sum (in order to retain feasibility of the budget constraints) while increasing a proxy function that matches the LP objective on integer points and remains boundedly close to the LP objective on fractional points. This is repeated until at most a few fractional variables remain, and the effect of some final required roundings can be bounded against the value of an initial partial-enumeration phase. Since the final solution is obtained by a series of increasing steps for the proxy function, it will have high value compared to the original LP solution (for the correct partially-enumerated set). A technical point for the analysis is that a series of such integer solutions must be produced so that the effect of the final required roundings are small. Some simple alterations of this analysis will also yield results for single and k-stage versions as well as for a version in which the first and second stage budgets are specified in the input.

Proof. (of Theorem 2.3.3) We formulate the following natural LP:

$$\begin{aligned}
\max \sum_{(i,v)} (p_i v_v) x_{iv} & \\
\text{s.t.} \sum_{e \in P(i,v)} y_e + \sum_{e \in P(i,v)} z_e^i \geq x_{iv} & \quad \forall (i,v) \\
\sum_e y_e c_e + \sum_e z_e^i (M^{ie} c_e) \leq B & \quad \forall i \\
x_{iv} \leq 1 & \quad \forall (i,v)
\end{aligned}$$

Here, p_i denotes the probability that node i is the ignition point under Π (the scenario where i is the ignition point is *scenario* i). In the associated IP, x_{iv} is 1 if node v is protected in scenario i, and 0 otherwise. Also, y_e is 1 if edge e is bought in the first stage, and 0 otherwise, and z_e^i is 1 if edge e is bought in the second stage for scenario i, and 0 otherwise. Constraints of the first form capture that if node v is protected in scenario i then it must be that some edge on the path from i to v is purchased either in the first stage or in the second stage for scenario i. Constraints of the second form capture that at most B can be spent buying edges in scenario i over the first and second stages combined. Preprocess by setting y_e to 0 if $c_e > B$, and z_e^i to 0 if $M^{ie} c_e > B$: the optimal solution can not use these options. Let δ denote the diameter of the tree.

Notice that in this LP, given a set of y_e and z_e^i, we can automatically determine the best x_{iv}. Following [1] we rewrite the problem as the following nonlinear optimization problem:

$$\max L(x) = \sum_{(i,v)} (p_i v_v) \min\{1, \sum_{e \in P(i,v)} y_e + \sum_{e \in P(i,v)} z_e^i\}$$

$$\text{s.t.} \sum_e y_e c_e + \sum_e z_e^i (M^{ie} c_e) \le B \quad \forall i, \text{ and } x_{iv} \le 1 \quad \forall (i,v).$$

Consider the function: $F(x) = \sum_{(i,v)} (p_i v_v) \Big[1 - \Big(\prod_{e \in P(i,v)} (1 - y_e) \Big) \Big(\prod_{e \in P(i,v)} (1 - z_e^i) \Big) \Big]$.

Lemma 2.3.4. *$F(x)$ has the following key properties:*

1. *$F(x)$ coincides with $L(x)$ when all the y_e and z_e^i are integral.*
2. *On non-integral (y_e, z_e^i) vectors, $F(x)$ is at least $(1 - (1 - \frac{1}{2\delta})^{2\delta}) L(x)$.*
3. *$F(x)$ is concave in the direction of a vector that changes at most 2 y_e values at a time and changes no $z_{e'}^i$ values. $F(x)$ is concave in the direction of a vector that changes at most 2 z_e^i values for a common i at a time and changes no y_e values, and changes no $z_e^{i'}$ values for $i' \neq i$. Based on the budget constraint coefficients of the changing variables, vectors of this type can be found through appropriate scaling that maintain all budget constraints.*
4. *Let Y, Z denote sets corresponding to the y_e, z_e^i decision variables being set to 1. $F(X)$ defined on subsets of $Y \cup Z$ is a submodular set function.*

Properties 1, 2 and 4 hold just as in [1] since the function $F(x)$ has the same form (though now there is a formal distinction between first and second stage variables). For property 3: the number of terms in F's product which change for any particular (i, v) is at most 2: concavity results as in [1], but unlike in [1], not any set of two fractional decision variables will maintain budget feasibility).

Denote by LP$[I_0, I_1]$ the original LP (post preprocessing) subject to the additional constraints that decision variables in I_1 are set to 1 and decision variables in I_0 are set to 0. We use an auxiliary algorithm $\mathcal{A}$ identical to [1] except for a key additional point. First, $\mathcal{A}$ computes the optimal solution x^{LP} to LP[I_0, I_1] by some known polynomial-time algorithm, then $\mathcal{A}$ transforms this solution into $x^{\mathcal{A}}$ by a series of pipage steps. Each pipage step is as follows. If there exists only a single fractional variable among the y_e, and for every i there is at most a single fractional variable among the z^i_e, stop. Otherwise, select either two fractional y_e or two fractional z^i_e for a common i and consider the vector that maintains all budget constraints as one is increased while the other is decreased: this vector intersects the boundary of the feasibility polytope at two points. At one point the first decision variable has become 0 and the second has become 1, at the other point the second decision variable has become 0 and the first has become 1. Both points are feasible since all budget constraints are maintained, and one has $F(X)$ at least as great as the previous solution due to the concavity of F along the vector. We replace the current solution with this higher-$F(X)$ solution that has a greater number of integral variables.

Each pipage step of $\mathcal{A}$ reduces the number of fractional components of the current vector. Finally $\mathcal{A}$ outputs an *almost-integral* feasible vector $x^{\mathcal{A}}$ which has at most one fractional first-stage variable, and at most one fractional second-stage variable for each scenario i.

As in [1], this rounding procedure gives $F(\mathcal{A}) \geq F(x^{LP})$. Defining $J_1 = \{(i,v) : i \text{ is separated from } v \text{ by } I_1\}$, and from property 2 of the lemma:

$$F(x^{LP}) \geq \sum_{(i,v)\in J_1} p_i v_v + \left(1-(1-\frac{1}{2\delta})\right)^{2\delta} \sum_{(i,v)\in J\setminus J_1} (p_i v_v) \min\{1, \sum_{e\in P(i,v)} (y_e)^{LP} + \sum_{e\in P(i,v)} (z^i_e)^{LP}\}$$

Now we prove that this algorithm (see boxed description on the following page) meets claim of Theorem 2.3.3. First observe that the algorithm spends at most $2B$ for scenario i: pipage rounding maintains budget feasibility for every scenario and the final roundings used to achieve integrality round up at most a single fractional decision variable per scenario. Our preprocessing guarantees that this single round up costs at most B in addition to the cost of the fractional solution returned by $\mathcal{A}$.

Let X^* be the optimal set of decision variables, let Y^* denote the first stage variables in X^*. If $|Y^*| \leq 3$, then step 0. finds a $(1+\epsilon)$ approximation to OPT. So, we address the case when $|Y^*| \geq 4$. W.l.o.g. we can assume that the set of decision variables is ordered such that $Y^* = \{1, ..., |Y^*|\}$ and for each $i \in Y^*$, among the elements $\{i, ..., |Y^*|\}$ the element i protects the maximum total weight of (i, v) pairs which are not already protected by the set $\{1, ..., i-1\}$.

For the iteration in which $I_1 = \{1, 2, 3, 4\}$, let q denote the number of runs of the while loop. Since each run of the while loop either terminates the iteration or sets a first stage variable to 0, q is at most $n - 4$. During the iteration the algorithm finds a series of q feasible solutions to the LP. Let I_0^j denote I_0 in the jth run of the while loop. The jth feasible solution $\hat{X}_j$ has $\hat{X}_j \cap I_0^j = \emptyset$ (from the form of the algorithm). Index the elements of I_0^q in the order that the algorithm adds them to I_0, that is, $I_0^j = \{i_1, ..., i_j\}$ where i_l is the index of the lth first stage variable added to I_0 for this iteration.

The main algorithm.

0. For each set of at most three y_e, set them to 1, then find the PTAS 2nd stage decision that can be made in each scenario (no additional first stage edges purchased), and evaluate the objective of each such solution. Take the best such solution and call it q^*.

1. For each $I_1 \subseteq Y$ such that $|I_1| = 4$ and $\sum_{i \in I_1} c_i \leq B$:

- Set $I_0 = \emptyset$.
- Set $t = 0$.
- While $t = 0$: apply $\mathcal{A}$ to LP$[I_0, I_1]$.
    1. If all the $x_i^{\mathcal{A}}$ (decision variables in either stage) are integral, then set t to 1 and set $\hat{x}$ to $x_i^{\mathcal{A}}$.
    2. Else, if $x_i^{\mathcal{A}}$ has no fractional y_e, then round up any fractional z_e^i, set t to 1 and set $\hat{x}$ to $x_i^{\mathcal{A}}$ with the rounded up second stage variables.
    3. Else, if neither of these conditions holds, round down the single fractional y_e and round up all fractional z_e^i, set $\hat{x}$ to $x_i^{\mathcal{A}}$ with the rounded variables. Also, add the index of the y_e that was rounded down to I_0.
    4. If $F(\hat{x}) > F(\bar{x})$, then set $\bar{x}$ to $\hat{x}$. *(Since $\hat{x}$ and $\bar{x}$ are integral, this chooses the highest L-value among all the $\hat{x}$ considered by the algorithm).*

Assume first that $I_0^q \cap Y^* = \emptyset$. That is, when the iteration terminates, no first stage variables used by OPT have been forced to 0: OPT is a feasible solution for LP$[I_1, I_0^q]$. Since this is the last run of the while loop, it must have ended in an *if* statement of one of the first 2 types. In the first case: all the $x_i^{\mathcal{A}}$

(decision variables in either stage) are integral and $x_i^{\mathcal{A}}$ is the outcome of pipage rounding of the fractional optimal of LP[I_1, I_0^q]. In particular: since $\hat{x}$ is integral, $L(\hat{x}) = F(\hat{x}) = F(x_i^{\mathcal{A}}) \geq F(x^{LP})$. For the second case, rounding up the second-stage variables only increases the value of F, and after the rounding we have an integral solution, so $L(\hat{x}) = F(\hat{x}) \geq F(x_i^{\mathcal{A}}) \geq F(x^{LP})$. Either way, the following inequality derived from property 2 and the fact that OPT is feasible for LP[I_1, I_0^q] now give that $\hat{x}$ is a budget-balanced $(1-(1-\frac{1}{2\delta})^{2\delta})$-approximation:

$$\begin{aligned} F(x^{LP}) &\geq \sum_{(i,v)\in J_1} p_i v_v + \Big(1-(1-\frac{1}{2\delta})\Big)^{2\delta} \sum_{(i,v)\in J\setminus J_1} (p_i v_v) \min\{1, \sum_{e\in P(i,v)} (y_e)^{LP} + \sum_{e\in P(i,v)} (z_e^i)^{LP}\} \\ &\geq \Big(1-(1-\frac{1}{2\delta})\Big)^{2\delta} OPT. \end{aligned}$$

Now, assume that $I_0^q \cap Y^* \neq \emptyset$. Let I_0^{s+1} be the first I_0 in the series $I_0^1, ..., I_0^q$ that has nonempty intersection with Y^*: the sth run of the while loop is the first run of the while loop for this iteration in which the algorithm adds a first stage variable from Y^* to I_0 (call that variable i_s). The algorithm adds i_s to I_0 after considering a solution $\hat{x}$ in which i_s was the single fractional first stage variable was rounded down (this is the third type of *if* statement in the while loop). We claim that the $\hat{x}$ that resulted when i_s was rounded down (and fractional second stage variables were rounded up) was a $(1-(1-1/2\delta)^{2\delta})$-approximation. Proving this claim will be establish Theorem 2.3.3.

As in [1], $F(X)$ defined on subsets of $Y \cup Z$ is a submodular set function. Thus, we have the *diminishing-returns* property: for any subsets R and G of $Y \cup Z$ and any element $i \in Y \cup Z$, we get $F(R \cup i) - F(R) \geq F(R \cup G \cup i) - F(R \cup G)$. Now, letting h denote a member of Y^* which is not in $\{1, 2, 3, 4\}$, and letting H

denote any superset of $\{1,2,3,4\}$:

$$\begin{aligned}
1/4F(I_1) = 1/4F(1,2,3,4) &= 1/4[F(\{1,2,3,4\}) - F(\{1,2,3\})+ \\
&\qquad F(\{1,2,3\}) - F(\{1,2\})+ \\
&\qquad F(\{1,2\}) - F(\{1\})+ \\
&\qquad F(\{1\}) - F(\emptyset)] \\
&\geq 1/4[F(\{1,2,3,h\}) - F(\{1,2,3\})+ \\
&\qquad F(\{1,2,h\}) - F(\{1,2\})+ \\
&\qquad F(\{1,h\}) - F(\{1\})+ \\
&\qquad F(\{h\}) - F(\emptyset)] \\
&\geq F(H \cup \{h\}) - F(H).
\end{aligned}$$

The first equality results from a collapsing sum where we remove the final $+F(\emptyset)$ since it is 0 (since the tree is connected and every scenario has a source). By the labeling of the decision variables in Y^*: since h is not in $\{1,2,3,4\}$, the additional marginal value h protects beyond what is protected by any prefix of $\{1,2,3,4\}$ is at most the additional value that the index which does follow the prefix protects. Finally, we apply the diminishing-returns property 4 times to get the final inequality.

Also, as in [1], rounding up a fractional solution produced by $\mathcal{A}$ only increases the value of F. Let $x^{\mathcal{A}}$ denote the unrounded solution returned by $\mathcal{A}$. Let $I(x^{\mathcal{A}})$ be the integral positive elements of $x^{\mathcal{A}}$, let $\{j_1,...,j_i\}$ denote the set of fractional second stage variables in $x^{\mathcal{A}}$, and i_s denote the fractional first stage variable in $x^{\mathcal{A}}$ from Y^*. Then $\hat{x}$ is $I(x^{\mathcal{A}}) \cup \{j_1,...,j_i\}$, so we can use the integrality of $\hat{x}$ to bound its LP value as follows:

$$L(\hat{x}) = L(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\}) = F(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\})$$

Adding and subtracting a common quantity:

$$= F(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\} \cup \{i_s\}) - \\ \Big(\underbrace{F(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\} \cup \{i_s\}) - F(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\})}\Big)$$

Apply bound to bracketed quantity since $I(x^{\mathcal{A}})$ contains $\{1, 2, 3, 4\}$ and $i_s \in Y^*$:

$$\geq F(I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\} \cup \{i_s\}) - 1/4F(I_1) \geq F(x^{\mathcal{A}}) - 1/4F(I_1)$$

The second inequality holds because F increases when its argument is rounded up, and $I(x^{\mathcal{A}}) \cup \{j_1, ..., j_i\} \cup \{i_s\}$ is just $x^{\mathcal{A}}$ rounded up. Now write out $F(x^{\mathcal{A}})$:

$$= \sum_{(i,v)\in J_1} p_i v_v + \sum_{(i,v)\in J\setminus J_1} (p_i v_v)\Big[1 - \Big(\prod_{e\in P(i,v)} (1 - (y_e)^{\mathcal{A}})\Big)\Big(\prod_{e\in P(i,v)} (1 - (z_e^i)^{\mathcal{A}})\Big)\Big] - \\ 1/4F(I_1)$$

$$= 3/4 \sum_{(i,v)\in J_1} p_i v_v + \sum_{(i,v)\in J\setminus J_1} (p_i v_v)\Big[1 - \Big(\prod_{e\in P(i,v)} (1 - (y_e)^{\mathcal{A}})\Big)\Big(\prod_{e\in P(i,v)} (1 - (z_e^i)^{\mathcal{A}})\Big)\Big]$$

Pipage rounding produces $x^{\mathcal{A}}$ from x^{LP} while increasing F:

$$\geq 3/4 \sum_{(i,v)\in J_1} p_i v_v + \\ \sum_{(i,v)\in J\setminus J_1} (p_i v_v)\Big[1 - \Big(\prod_{e\in P(i,v)} (1 - (y_e)^{LP})\Big)\Big(\prod_{e\in P(i,v)} (1 - (z_e^i)^{LP})\Big)\Big]$$

Apply the well-known inequality which holds for all fractional solutions:

$$\geq 3/4 \sum_{(i,v)\in J_1} p_i v_v + \\ (1 - (1 - 1/2\delta)^{2\delta}) \sum_{(i,v)\in J\setminus J_1} (p_i v_v) \min\{1, \sum_{e\in P(i,v)} (y_e)^{LP} + \sum_{e\in P(i,v)} (z_e^i)^{LP}\}$$

Notice that $3/4 \geq (1 - (1 - 1/2\delta)^{2\delta})$. Also, x^{LP} is the optimal solution for

LP[I_1, I_0^s] and X^* is feasible for LP[I_1, I_0^s]. Thus, the last quantity is bounded below by $(1-(1-1/2\delta)^{2\delta})L(X^*) = (1-(1-1/2\delta)^{2\delta})OPT$. □

Suppose that the division of the budget between first and second stages is specified in the input as (B_1, B_2). Adding the additional constraints $\sum_e y_e c_e \leq B_1$ and $\sum_e z_e^i(M^{ie}c_e) \leq B_2$ for all i to the LP alters our analysis only slightly: preprocess to eliminate decision variables that are too expensive to fully buy in their corresponding stages, the algorithm now enumerates over four-member sets of first-stage decision variables, at the conclusion of the pipage phase the remaining fractional first-stage variable is rounded down (so B_1 is respected) and at most one second-stage variable per scenario is rounded up (B_2 is overspent by at most a factor of 2), first stage variables which are rounded down are excluded one by one in the iterations of the while loop. Thus, we get:

Theorem 2.3.5. *Given a specific first-stage budget B_1 and second-stage budget B_2, there exists a $(1-(1-\frac{1}{2\delta})^{2\delta})$-approximation algorithm for the 2-stage stochastic GPP in trees that respects B_1 and violates B_2 by a factor of at most 2 (each ignition set has size 1, δ denotes the diameter of the tree).*

Stochastic single-stage and k-stage results. In the limiting single-stage stochastic case (where second-stage action is prohibitively expensive) there is only a single budget constraint: the proof of Theorem 2.3.3 can be simplified so that it directly follows [1] to get:

Theorem 2.3.6. *There exists a $(1-(1-\frac{1}{\delta})^{\delta})$-approximation algorithm for the single-stage stochastic GPP in trees provided that each ignition set has size 1 (δ denotes the diameter of the tree).*

We mention several corollaries of Theorem 2.3.6. First, Theorem 2.3.6 extends to a $(1 - (1 - \frac{1}{\delta})^{\delta}, O(\log n))$ bicriteria approximation for general graphs. For the 2-stage stochastic GPP in trees with single ignition node, consider the algorithm that chooses the better performance between spending all of B in stage 1 vs. spending all of B in stage 2: apply Theorem 2.3.6 for stage 1 and the PTAS for deterministic single-source GPP for stage 2 assuming that the optimal solution earns α(OPT) in the first stage, and minimize over $\alpha \in (0, 1)$ to get a worst case guarantee of $(0.387, 1)$. For a constant number of scenarios, use Theorem 2.3.2 in the place of Theorem 2.3.6 to get a $(.5(1-\epsilon), 1)$- approximation.

A cautionary note: for the 2-stage stochastic GPP in trees with single ignition node, consider a method that treats the stages independently by spending B according to the method from Theorem 2.3.6 in the first stage, and then, once the ignition point is realized, spending an additional B according to the method from Theorem 2.3.1. That is, it matches the spending level of Theorem 2.3.3. It is tempting to hope that this approach (which ignores coordination between first- and second-stage edge removals) will protect at least $(1 - \frac{1}{e})$ of the optimal value (since each method protects at least this fraction within the stage where it is executed). Unfortunately, this is not the case: due to the nonuniform inflation factors allowed for edge costs, edge removals which appear to yield modest returns may be overlooked in the first stage, but become prohibitively expensive in the second stage (that is, they may cost more than B). In Figure 2.2 we give an example in which even the ability to solve each 1-stage problem optimally can't protect $(\frac{1}{2} + \epsilon)$ of the optimal value for $\epsilon > 0$.

The k-stage stochastic graph protection problem in trees (for constant k) has

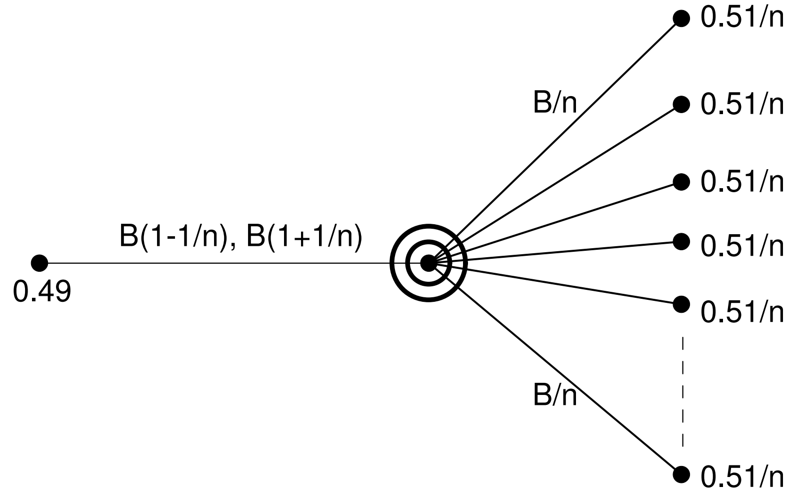

Figure 2.2: **Treating stages independently gives weaker performance:** The central circled node has value 1, other nodes have value 0. The left node has ignition probability 0.49, each right node has ignition probability $0.51/n$. Edges on the right cost B/n in both stages. The left edge costs $B(1-\frac{1}{n})$ in the first stage and $B(1+\frac{1}{n})$ in the second stage. The optimal solution removes the right edge in stage 1 and the relevant left edge in stage 2 (protecting value 1). For sufficiently large n, spending B in the first stage optimally removes all of the edges on the right, protecting value 0.51 in stage 1 (in stage 2 no additional value can be protected). Replacing 0.49 by $(0.5-\epsilon)$, and 0.51 by $(0.5+\epsilon)$: no more than $1/2$ of the optimal can be protected when the first stage edge-removals ignore the second-stage options. Notice: the difference in inflation factors can be arbitrarily close to 1.

k stages in which information is revealed and decisions about edge removal are made (rather than one or two stages). This information can be considered as updates that arrive at k specific times which condition the distribution on where the ignition will occur (by specifying that the ignition will occur among some particular subset of the nodes). For each stage the input includes a partition of the node set, and the partition for stage i refines the partition for stage $i-1$. In each stage the planner has the option to remove additional edges from the graph at some (stage, partition piece)-specific cost. A solution specifies which edges will be removed for each partition piece realization at each stage. The total cost incurred for each realized sequence of k partition pieces should be B.

Theorem 2.3.7. *For a restricted class of information revelation hierarchies, there exists a bicriteria $(1-(1-\frac{1}{k\delta})^{k\delta}), 2+\epsilon)$-approximation algorithm for the k-stage stochastic GPP in trees provided that each ignition set has size 1 (k is a constant, δ denotes the diameter of the tree).*

Theorem 2.3.7 requires that the number of partition pieces added over all stages excluding the last stage (in which any of n points may be realized) is bounded by a constant: guessing the optimal division of the budget to ϵ/k-precision for each possible information realization takes polynomial time. As in the (B_1, B_2) case: impose additional constraints based on the guess of optimal budget division, reject too-expensive decision variables, pipage round (now roundings take place between pairs of fractional variables that correspond to a common partition piece within a stage). Last, round up all fractional variables (see Section 2.8.2 for details).

If there is a specified budget for each of the k stages, then the guessing (enumeration) may be dropped: with no requirements on the information revelation hierarchy the same analysis gives a $(1-(1-\frac{1}{k\delta})^{k\delta})$ value-protection guarantee which violates each stage's budget by a factor of at most 2.

2.4 Reductions to, and Extensions of, Related Problems

A looser $(1 - \frac{1}{e})$ guarantee which matches Theorem 2.3.6 asymptotically may be obtained by reducing single-stage stochastic GPP in trees to the weighted Budgeted Separating Minimum Cut Problem in trees for which the analysis of Engelberg, et. al [15] applies: submodularity of the objective allows application of the result of Sviridenko [48]). The tighter result in Theorem 2.3.6 can alternately be proved by a more subtle reduction to MCKP addressed in [1].

Reducing wBSMC in trees to MCKP gives the tightest existing result for wBSMC in trees, beating both the reduction to submodularity in [15] and the weaker primal-dual $(\frac{1}{3})$-approximation in trees also given in [15]. The advantage of this tighter guarantee is substantial in trees of constant diameter. We describe these related problems and their stochastic multistage extensions below, then provide reductions mentioned above, and extensions of our results for GPP.

Maximum Coverage with a Knapsack Constraint (MCKP): Given a family $F = \{S_j : j \in J\}$ of subsets of a set $I = \{1, 2, ..., n\}$ with associated nonnegative weights w_j and costs c_j of the elements, and positive integer B, find a subset $X \subseteq I$ with $\sum_{j \in X} c_j \leq B$ so as to maximize the total weight of the sets in F having nonnegative intersections with X. (This version of the problem is as stated in [1]).

- *Stochastic MCKP:* There is also a distribution Π: each scenario specifies how much value will be received for covering the subset S_j for each j. The objective is to maximize the expected weight of subsets covered.

- *Multistage MCKP:* Elements may be purchased in different stages at a cost that is stage-, scenario-, and element-dependent (costs are specified in the input). Stochastic multistage versions of wBSMC in trees reduce to these MCKP problems.

weighted Budgeted Separating Multiway Cut (wBSMC) in a Tree: Call the tree $T = (V, E)$. There is a weight function on pairs of nodes $\omega : V \times V \to \mathbb{Z}^+$, and a budget B: find the subset of edges $C \subseteq E$ of cost at most B whose separation weight is maximized.

- *Stochastic wBSMC in Trees:* In addition to the input for deterministic wBSMC in trees, there is also a distribution Π. Each scenario in the distribution specifies how much value will be received for pair (u, v) if u and v are separated by the solution. The objective is to maximize the expected weight of pairs separated.
- *Multistage:* An edge may be purchased in different stages at a cost that is stage, scenario, and edge dependent. These costs are specified in the input.

Reductions.

Proof. (stochastic single-source GPP in trees reduces in polynomial time to wBSMC in tree graphs): The tree is the same. For each (ignition node, node) pair (i, v) create a (node, node) pair (i, v) of weight $w_{iv} = p_i v_v$. The budget B is the budget is the budget for wBSMC.

Suppose that Y is a set of edges of cost at most B. Then the set $X \subseteq I$ of edges in the wBSMC instance corresponding to edges in Y costs the same amount (at most B). The set Y protects v in scenario i iff Y contains an edge on the path from v to ignition point i iff X contains an edge on the path from v to i iff X gets value w_{iv} for separating i and v. The weight of pairs separated by X is exactly the expected value protected by Y (since for Y the expected value protected can be written as the sum over (node, scenario)-pairs). Instead, starting from a budget-balanced set of edges X for wBSMC, we can use the same equivalences to produce a budget-balanced set of edges which protects expected value equal to the weight of pairs separated by X. □

Proof. (stochastic single-source GPP in trees reduces in polynomial time to MCKP): The set I of ground elements contains an element corresponding to each edge of the tree. The element corresponding to edge e has cost c_e. For each (ignition point, node) pair (i, v) we create a subset S_{iv} consisting of elements corresponding to edges on the path between i and v. The weight of S_{iv} is $p_i v_v$. The budget B is the budget is the budget for MCKP.

Suppose that Y is a set of edges of cost at most B. Then the set $X \subseteq I$ of elements corresponding to edges in Y costs the same amount (at most B). The set Y protects v in scenario i iff Y contains an edge on the path from v to ignition point i iff X contains an element in S_{iv} iff X covers S_{iv} and gets value $p_i v_v$. The weight of subsets covered by X is exactly the expected value protected by Y (since for Y the expected value protected can be written as the sum over (node, scenario)-pairs). Instead, starting from a budget-balanced set of elements X, we can use the same equivalences to produce a budget-balanced set of edges which protects expected value equal to the weight covered by X. □

Proof. (wBSMC in trees reduces in polynomial time to MCKP): The set I of ground elements contains an element corresponding to each edge of the tree. The element corresponding to edge e has cost c_e. For each (node, node) pair (i, v) we create a subset S_{iv} consisting of elements corresponding to edges on the path between i and v. The weight of S_{iv} is w_{iv}. The budget B is the budget is the budget for MCKP.

Suppose that Y is a set of edges of cost at most B. Then the set $X \subseteq I$ of elements corresponding to edges in Y costs the same amount (at most B). The set Y separates v from i iff Y contains an edge on the path from v to i iff X contains an element in S_{iv} iff X covers S_{iv} and gets value w_{iv}. The weight of subsets covered by X is exactly the weight separated by Y (since for Y the weight separated can be written as the sum over (node, node)-pairs). Instead, starting from a budget-balanced set of elements X, we can use the same equivalences to produce a budget-balanced set of edges which separates pairs of total weight equal to the weight covered by X. □

Extensions.

The features of the LP we analyzed (objective function and budget constraints) also hold for the natural LPs for these problems: the analysis proving Theorems 2.3.3, 2.3.5, 2.3.6, and 2.3.7 can be extended with identical guarantees to the corresponding multistage stochastic MCKP and wBSMC-in-trees generalizations.

We explain the MCKP case below.

In the 2-stage stochastic MCKP, there are a set of ground elements each with a given first-stage cost, and a group of subsets $S_1, S_2, ..., S_j$. Also the input includes a distribution Π over scenarios. Each scenario has a specific probability and specifies a value for each S_i that will be awarded if at least one element of S_i is purchased by the solution. Also, each scenario has a second-stage cost for each element. The problem is to find first-stage element purchases and second-stage element purchases for each scenario (of total cost at most B) that maximize the expected value awarded when a scenario is realized from the distribution.

In our analysis of Theorems 2.3.3, 2.3.5, 2.3.6, and 2.3.7, we haven't used the form of the tree (except for the existence of unique paths in which a single edge added to the solution protects the node from an ignition point), or that the costs in the second stage are larger than the costs in the first stage, or that the values of the nodes were uniform across scenarios. The LP that we describe below represents the 2-stage stochastic MCKP, and our method works in the same way with two times the cardinality of the largest S_i replacing the diameter of the tree in the description for stochastic graph protection. Notice that the form of F and L are unchanged, and that arguments about submodularity of F are unchanged.

Let x_{iq} be a decision variable that is 1 if some element of set S_q is purchased in either the first stage or the second stage of scenario i and 0 otherwise. Now decision variables y_e, z_e^i correspond to the first and second-stage purchases of element e.

$$
\begin{aligned}
\max \sum_{(i,q)} (p_i v_i(S_q)) x_{iq} & \\
s.t. & \\
& \sum_{e \in S_i} y_e + \sum_{e \in S_i} z_e^i \geq x_{iq} \quad \forall (i,q) \\
& \sum_{e} y_e c_e + \sum_{e} z_e^i (M^{ie} c_e) \leq B \quad \forall i \\
& x_{iq} \leq 1 \quad \forall (i,q)
\end{aligned}
$$

This linear-programming relaxation is identical to that in the proof of Theorem 2.3.3.

2.5 Tuning 2-Stage Input Form to Describe Spread Dynamics

In this section we comment on how the form of the input to the 2-stage GPP may be tuned to capture more subtle information about the dynamics and extent of the spread of a harmful process.

Consider a diffusive process in a graph that spreads according to some function across the edges of a graph until a process-ending event. The state of each node at the occurrence of the process-ending event is either that the node has been reached by the diffusive process or it has not. If no actions are taken to remove edges from the graph then for each scenario there will be a set of nodes that has been reached by the process, and we shall call this set the *affected envelope* of the scenario.

The form of the affected envelope for a scenario may depend on several different variables related to the scenario. For example, in a richer version of our fire application, each scenario may be composed of an ignition point, a sequence of wind directions each of which hold for a specified amount of time, and the amount of time before a fire ending event. That is, a scenario can be specified to be all the stochastic data that are used to model the final perimeter of the fire at termination. The affected envelope for a scenario is just the set of all nodes which burn (for example, in the Farsite simulation for the starting scenario data-Farsite is a well-known and widely-used fire simulation software [18]).

Another example of an *affected envelope* is directional spread in a tree: consider a river and stream network in which the introduction of an invasive species at a stochastic location results in the species spreading downstream until it encounters a kind of treatment zone (the treatment zone is the management action taken to prevent spread). Then the affected envelope for a scenario in which the introduction takes place at a particular node is the set of all nodes which are downstream from the introduction site.

Suppose that we have a polynomial-time way to compute the affected envelope for each scenario (as we do in both of the examples mentioned, namely, via Farsite, and simple depth first search when each edge in the stream system is given a specified direction).

Then create the input for the stochastic 2-stage GPP as follows. First, the scenario-dependent value of a node in the affected envelope will be equal to the value lost at that node if it is affected. The scenario-dependent value of a node not in the affected envelope will be 0. All edge costs prior to the final stage will be unchanged. The cost of an edge that has both endpoints in the affected envelope will also have unchanged cost in the final stage. The cost of an edge which does not have both endpoints in the affected envelope will now have final-stage cost 0.

Now, what does a set of edge treatments get credit for? In each scenario, a solution gets credit for protecting nodes which were threatened in that scenario (and does not get credit for protecting nodes which were not threatened in that scenario). How does a solution protect a node in a particular scenario? A solution protects a node exactly when it blocks every path *contained in the affected envelope* from the outbreak site to affected node. This is apparently different than how we have described our model before where to protect a node a solution had to block every path *contained in the graph* from the outbreak site to the node. The difference after the tuning is due to the fact that the paths contained in the graph but not in the affected envelope can be blocked for 0 cost in the final stage by every solution. This input tuning is a modeling device to remove the incentive to treat such edges associated with scenarios that do not actually use them: if an edge whose cost has been tuned to 0 in the final stage is chosen the treatment plan will not actually treat this edge in the final step.

This works for our applications because blocking paths never causes the affected envelope to expand to contain nodes that weren't in the affected envelope when no treatment was applied. This argument relies on the fact that adding treatment never *steers* the diffusive process into a region that it would not have visited in the untreated scenario. This fact will not hold for an invasive species whose spread is *active* in the sense that the species will spread greedily (and

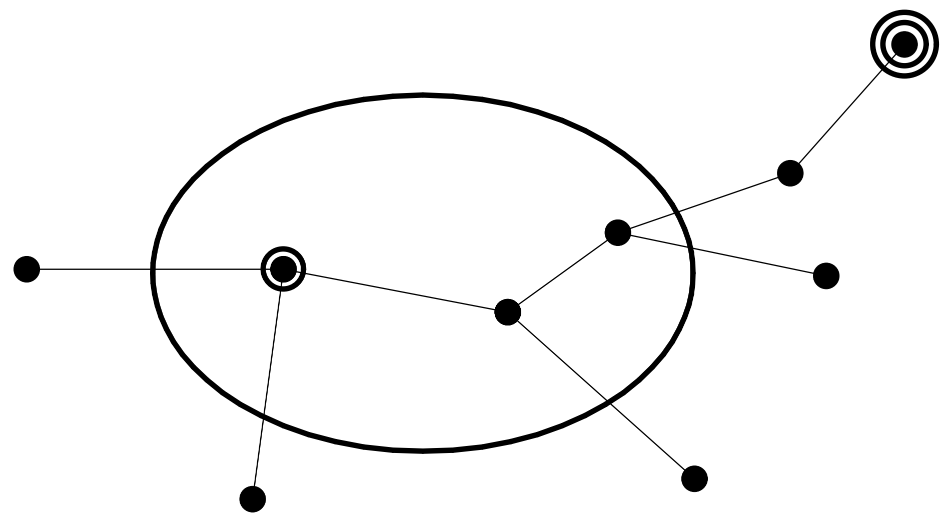

Figure 2.3: After tuning: protecting the double-circled node from the ignition node (indicated by the single-circled node) doesn't require buying an edge on the path between them (since that path is not entirely contained within the affected envelope).

lazily) towards the closest area where food, etc. is available, and thus will divert the course of its invasion if our management actions simply provide an alternate food path through the landscape.

In large landscapes with small fire events this approach for input tuning makes the guarantees of our algorithms much more meaningful.

2.6 Single-stage Extension to Probabilistic Edge Transmission

In ecological fact, fuel-treated areas are not 100% burn resistant (e.g. they may burn if extreme weather arises). Also, different types of treatments (with different costs) may reduce the probability of fire passing between adjacent parcels by different amounts. These considerations motivate a version of GPP in which the input specifies a more complicated relationship between spending on each edge and the resulting transmission probability across that edge. Previously we had two options: pay 100% of the edge cost to get probability of transmission 0, or pay 0% of the edge cost to get probability of transmission 1.

To single-stage stochastic GPP where each ignition set has size 1, we add the feature that each edge has (as part of the input) a specified monotonically-decreasing step function that gives the probability of transmission across that edge as a function of the spending level (the spending level may range from 0% to 100% of the edge cost, the events of transmissions across edges are assumed to be independent). We give an approximation result assuming that the running time of the algorithm is allowed to depend polynomially on the number of steps in each step function. The objective remains to maximize the expected value protected from the ignition point, only now this expectation is over realization of both the scenario and the individual edge-transmission events that arise.

The analogous notion for MCKP is of *probabilistic element failure*: for each element there is a step function that represents the probability that the element will *fail to cover the subsets which contain it* (generalizing that an element e fails to cover subsets which contain it with probability 1 if we do nothing, and with probability 0 if we pay c_e). The objective is to maximize the expected weight of subsets covered, where this expectation is over both element and scenario realization. The generalization of wBSMC in trees to a case with probabilistic edge occurrence reduces to MCKP with *probabilistic element failure*. This reduction

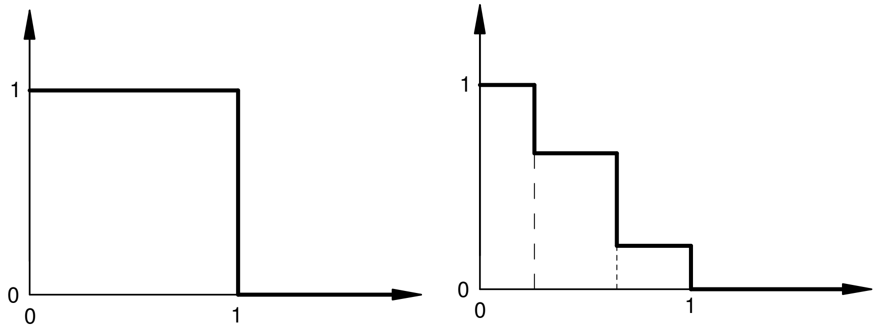

Figure 2.4: A step function gives the probability of transmission across that edge as a function of the spending level on that edge. The deterministic version addressed in previous sections is the 0-1 step function on the left.

is provided in Section 2.8.5 of the technical addendum.

Theorem 2.6.1. *There exists a $(1 - \frac{1}{e})$-approximation algorithm for the single-stage stochastic GPP in trees with probabilistic edge transmission (provided that each ignition set has size 1). For MCKP with probabilistic element failure: there exists a $(1 - \frac{1}{e})$-approximation algorithm.*

Proof. Each (spending level, edge) pair is an element the solution can buy with cost corresponding to the spending level times the edge cost (we only have elements corresponding to critical spending levels at which the transmission probability instantaneously drops). Let X denote the set of such elements. The expected value protected is a set function over these elements. Denote this function by E. We wish to maximize this set function by buying elements subject to a knapsack constraint: if we show that this set function is submodular, [48] will immediately yield a $(1 - \frac{1}{e})$-approximation that is budget-balanced (provided that we can compute in polynomial time the element which gives largest improvement). To prove submodularity we will establish the law of diminishing returns: for an arbitrary (spending level, edge) pair denoted by a, if $A \subseteq B \subseteq X$, then $E(A \cup a) - E(A) \geq E(B \cup a) - E(B)$.

Let the edge of the (spending level, edge) pair a be denoted by e. According to the step function for e, buying a results in some probability of transmission α_i. Before a is added, A contains some set of elements which affect the transmission probability on e, and B contains a superset of these elements. Thus the probability of transmission on e is (weakly) larger for the set A than for the set B. In both cases, when a is added to a set, the new probability of transmission on e is the minimum of α_i and the current probability of transmission on e. The gap is larger for A than for B. Let $\wp_e(\cdot)$ denote the probability of transmission on e as a function of the set of elements:

$$\wp_e(A) \geq \wp_e(B) \Rightarrow \wp_e(A) - \wp_e(A \cup a) \geq \wp_e(B) - \wp_e(B \cup a)$$

Next, focus on a particular (ignition point, node) pair (i, v). If the path from i to v does not contain e, then adding e does not change the (i, v)th term in the expression for expected value protected. If the path from i to v does contain e, for each non-e edge on this i to v path, the probability of transmission under A is at least the probability of transmission under B. Let $P(Q)$ denote the probability that every edge on the i to v path (excluding e) transmits under Q:

$$\begin{aligned}
P(A) \geq P(B) &\Rightarrow P(A)(\wp_e(A) - \wp_e(A \cup a)) \geq P(B)(\wp_e(B) - \wp_e(B \cup a)) \\
&\Rightarrow P(A)\wp_e(A) - P(A)\wp_e(A \cup a) \geq P(B)\wp_e(B) - P(B)\wp_e(B \cup a) \\
&\Rightarrow (1 - P(A)\wp_e(A \cup a)) - (1 - P(A)\wp_e(A)) \geq \\
&\qquad\qquad (1 - P(B)\wp_e(B \cup a)) - (1 - P(B)\wp_e(B)) \\
&\Rightarrow E(A \cup a) - E(A) \geq E(B \cup a) - E(B).
\end{aligned}$$

The third line compares the changes in probability that v is protected from i which result when a is added to A and when a is added to B. The final inequality follows from summing change in expected valued protected over (ignition point, node) pairs (including pairs for which the addition of e caused no change). This establishes submodularity. Computing the change in E resulting from the addition of a single element simply requires computing the product along the (ignition point, node) path twice for each (i, v) pair. This takes polynomial time for each of polynomially-many elements. □

(The analysis for the MCKP case is almost identical, see Section 2.8.4).

2.7 Extensions to General Graphs

Single-stage: *Theorems 2.3.1, 2.3.2, 2.3.6 can be extended to bicriteria approximations for general graphs: the guarantees on value protected (expected value protected) are identical, but the budget is violated by a* $O(\log n)$*-factor.*

Multi-stage: *Assume that the number of scenarios is bounded by a constant, and that within each scenario the ratio of the largest inflation factor to the smallest inflation factor is bounded by a constant. Then Theorems 2.3.3, 2.3.5, and 2.3.7 can be extended to general graphs: the guarantees on expected value protected are identical, but the budget(s) is violated by a $O(\log n)$-factor.*

Single-stage. As in Engelberg, et. al [15] we apply the result of Räcke [40] on cut-capacity approximation: approximate the costs graph by a distribution over tree embeddings (each tree has diameter at most n), solve a suitably modified instance in each tree, translate solutions back to the original graph, select the best solution. We rely heavily on the exposition of Räcke's results by Andersen and Feige [2].

There are 2 main points in the analysis of this method. First, for an arbitrary tree in the distribution, the amount of expected value protected in the tree when a particular set of edges is removed is exactly the amount of expected value protected in the original graph when the same partition of the node set is enforced by removing a minimal set of edges from the original graph (minimal in the sense that no subset of these edges would match the partition of the node set that occurred in the tree).

Second, the optimal set of edges to remove from the original graph has expected distortion $O(\log n)$ where the expectation is with respect to the distribution over tree embeddings. Thus, some particular tree T in the distribution must distort the cost of the optimal set of edges by at most $O(\log n)$. Furthermore, the distribution Räcke uses has polynomial size and can be found in polynomial time [40], [2].

Thus, solving stochastic GPP in each tree from the distribution with respect to a budget $B \cdot O(\log n)$ guarantees that the optimal value in one of the trees will be at least the optimal value in the original graph, so that a guarantee in trees on expected value protected will give exactly the same guarantee on expected value protected in the original graph. Finally, since the methods we apply in trees are budget-balanced, the cost approximation is $O(\log n)$.

Multi-stage. If the number of scenarios is bounded by a constant and within each scenario the ratio of the largest inflation factor to the smallest inflation factor is bounded by a constant, then Theorems 2.3.3, 2.3.5, and 2.3.7 can be extended to general graphs: the guarantees on expected value protected are identical (where δ is at most n), but the budget(s) is violated by a $O(\log n)$-factor. To apply the result of Räcke [40] to the multistage case we need that some tree produced by the cut-capacity approximation has $O(\log n)$-distortion *for the optimal solution in every scenario* (not just for a single set of edges purchased in the first stage). By requiring that the number of scenarios is constant, we can obtain this as follows. For a fixed scenario, apply the Markov inequality: the probability that the cost of the optimal solution for that scenario is distorted by more than an additional constant k^2-factor (beyond its expected $O(\log n)$ distortion) is at most $1/k^2$. Apply a union bound over the scenarios to find that there is a $(k-1)/k$ probability that in every scenario the optimal solution is distorted by no more than an additional k^2-factor. So, with $(k-1)/k$ probability, there exists a solution in some tree produced by Räcke's cut-capacity approximation which spends at most B times $O(\log n)$ under any scenario and saves as much as the optimal. With our method in trees this gives the bicriteria results claimed.

Note: the requirement that within each scenario the ratio of the largest inflation factor to the smallest inflation factor is bounded by a constant is a technical point used to bound the distortion of the optimal set of edges. For more details, see [46].

2.8 Technical Addendum

2.8.1 A direct Polynomial-time Approximation Scheme for single-stage deterministic GPP with general ignition set

Claim: *A PTAS as asserted in Theorem 2.3.1 for tree graphs can be produced directly by extending the classic dynamic-programming (DP) framework for Knapsack.*

Proof. Suppose that we replace the single ignition node s, by an ignition set S.

The input to the problem is a connected tree $T = (V, E)$ with a specified ignition source set S, a non-negative *value* function $v : V \to \mathbb{Z}$, a non-negative *cost* function $c : E \to \mathbb{Z}$, and a *budget* B. Now the objective is to find the set of edges of cost at most B such that when the set is removed from T the value of nodes not reachable from any node in S is maximized. Notice that we can contract all edges with costs strictly greater than B since they will not be in any optimal solution.

First, break the tree into *components* by severing the tree at ignition nodes. That is, each *component* has the following property: for any node in that component, the component also contains all nodes that can be reached from that node by paths in the tree that do not contain ignition nodes except as endpoints. So ignition nodes will appear in multiple components (one appearance for each edge adjacent to the ignition node).

Now observe that in a particular component (which is itself a tree), all ignition points are leaves. We will run a Dynamic Programming method on each such component and then combine the resulting solutions using the observation that a non-ignition node v is protected in the original tree by a particular edge set exactly when v is protected in the component containing it by that edge set. Components that contain only nodes which are ignition nodes are treated separately since there is always an optimal solution that does not buy any edges in such components.

From the component Q choose an arbitrary non-ignition node to be the root r. For a pair of nodes $x, y \in V$, we will say y *is below* x if the shortest root to y path contains x. Further, if y is *below* x and is contained in an edge with x we will say that y is a *child* of x, The set of all *children of* x will be denoted $\chi(x)$. We will call the first edge on the shortest path from x to the root the *parent edge* of x, denoted e_x, and will refer to all edges and nodes in the component containing x in $T \setminus e_x$ as being *below* e_x (with e_x also being *weakly below* e_x).

For ease in analysis later, we describe the algorithm as having two stages: a first preprocessing stage which determines an ordering of the nodes of Q, and a main stage in which this ordering is used to run a dynamic program.

In the preprocessing stage, the algorithm finds an ordering of the nodes as follows:

1. Set $i = 1$.
2. Choose a non-root leaf of Q, label it node i.
3. Delete node i from Q.
4. $i \leftarrow i + 1$.
5. While $Q \setminus r$ is nonempty, go to step 2.
6. label r as node $n = |V(Q)|$.

Having produced this ordering, we are ready to run the main stage of the algorithm on the original Q.

Notice that under any solution each particular node is either unsafe or safe (that is, the node is either reachable from an ignition node or not).

During the main stage of the algorithm we maintain two array entries $A_s(j)$ and $A_u(j)$ for $j = \{1, ..., n = |V|\}$ (the label j refers to a node as labeled in the preprocessing stage). Each entry $A_s(j)$ is a list of pairs (t, w). A pair (t, w) in the list of entry $A_s(j)$ indicates that there is a set R of edges strictly below j which has cost exactly $t \leq B$, protects value exactly w weakly below j, and which is compatible with node j being safe. We say that a set of edges below node j is compatible with j being safe if the set of edges protects j from ignition from below, and j is specified to be safe.

Similarly, each entry $A_u(j)$ is a list of pairs (t, w). A pair (t, w) in the list of entry $A_u(j)$ indicates that there is a set R of edges below j which has cost exactly $t \leq B$, protects value exactly w weakly below j, and j is specified to be unsafe.

Notice that the ordering from the preprocessing stage provides a natural ranking of the children of a node j. Also, for each node j which is not a leaf in T the algorithm maintains two separate arrays of length $|\chi(j)|$ which consist of entries $g_s(j,1), g_s(j,2), ..., g_s(j, |\chi(j)|)$ and of entries $g_u(j,1), g_u(j,2), ..., g_u(j, |\chi(j)|)$.

Each entry is a list of pairs (r, u). A pair (r, u) in the list of entry $g_s(j, i)$ indicates that there is a set R of edges in which every edge is weakly below the parent edge of one of the first i children of node j, such that the total cost of R is exactly $r \leq B$, the total value protected by R weakly below one of the first i children of j is exactly u, and which is compatible with node j being safe from ignition traveling through one of j's first i children. We say that a set of edges below node j is compatible with j being safe (from ignition traveling through one of its first i children) if the set of edges protects j from ignitions below one of its first i children, and j is specified to be safe.

A pair (r, u) in the list of entry $g_u(j, i)$ indicates that there is a set R of edges in which every edge is weakly below the parent edge of one of the first i children of node j, such that the total cost of R is exactly $r \leq B$, the total value protected by R weakly below one of the first i children of j is exactly u, and j is specified to be unsafe.

For all types of arrays described, the lists do not contain all such pairs, but only keep track of the most efficient ones. To formalize this notion of efficiency we define the notion of one pair *dominating* another pair: a pair (t', w') *dominates* a pair (t, w) if $t' \leq t$ and $w' \geq w$, that is, if the set indicated by pair (t', w') costs at most as much and protects at least as much as the set corresponding to pair (t, w). Notice that we only compare pairs within a common array entry by this notion of domination: the status of j being safe (or unsafe) is uniform between any two pairs we shall make this comparison between.

Observe: Any set of edges which contains an edge e between two nodes of the same status (either both nodes are safe, or both nodes are unsafe) corresponds to a pair which is dominated by the pair corresponding to the set in which e has been removed.

Notice that domination is transitive. We will ensure that in any list no pair dominates another pair. This means that in any single list there can be at most one pair with cost $t \leq B$, and at most one pair with value w.

Consider the original tree. Each node can either be saved by cutting B worth of edges or it can not (this can be efficiently determined by performing a mincut computation in which the node is the source and there is a super-sink that has edges to every ignition node of cost $2B$). Find the largest-value single node saved by any B cost set, and call its value v^*. Notice that OPT is at least v^* and at most nv^*. Thus, since all costs and values are integers, each list has length at most $\min(B+1, nv^*+1)$. Now we are ready to give the dynamic program which constructs the lists.

```
If node 1 is an ignition node, set $A_u(1) \leftarrow \{(0,0)\}$, and $A_s(1) \leftarrow \emptyset$.
If node 1 is not an ignition node, set $A_u(1) \leftarrow \{(0,0)\}$, and $A_s(1) \leftarrow \{(v(1),0)\}$.
For $j \leftarrow 2$ to $n$
   If $j$ is a leaf in $Q$,
      If node j is an ignition node,
         set $A_u(j) \leftarrow \{(0,0)\}$, and $A_s(1) \leftarrow \emptyset$.
      If node j is not an ignition node,
         set $A_u(j) \leftarrow \{(0,0)\}$, and $A_s(1) \leftarrow \{(v(j),0)\}$.
   Else
      Let $y_1, y_2, ..., y_{|\chi(j)|}$ correspond to the natural ordering on the
       labels of the children of $j$.
      Set $g_s(j,1) \leftarrow A_s(y_1) \cup \{(r, u + c(e_{y_i})) : (r,u) \in A_u(y_1)\}$.
      Set $g_u(j,1) \leftarrow A_u(y_1) \cup \{(r, u + c(e_{y_i})) : (r,u) \in A_s(y_1)\}$

         For $i \leftarrow 2$ to $|\chi(j)|$
            $g_s(j,i) \leftarrow \{(r+t, u+w) : (r,u) \in g_s(y_{i-1}), (t,w) \in A_s(y_i)\} \cup$
               $\{(r+t, u+w+c(e_{y_i})) : (r,u) \in g_s(j, y_{i-1}), (t,w) \in A_u(y_i)\}$.
            $g_u(j,i) \leftarrow \{r+t, u+w) : (r,u) \in g_u(j, y_{i-1}), (t,w) \in A_u(y_i)\} \cup$
               $\{r+t, u+w+c(e_{y_i})) : (r,u) \in g_u(j, y_{i-1}), (t,w) \in A_s(y_i)\}$.
            Remove all dominated pairs from $g_s(j,i)$, as well as any pairs
               with cost $> B$.
            Remove all dominated pairs from $g_u(j,i)$, as well as any pairs
               with cost $> B$.
```

```
Set $A_s(j) \leftarrow \{(r + v(j), u) : (r, u) \in g_s(j, |\chi(j)|)\}$
Set $A_u(j) \leftarrow g_u(j, |\chi(j)|)$
Remove any dominated pairs from $A_s(j)$.
Remove any dominated pairs from $A_u(j)$.
If $j = |Q|$, for each spending amount of at most $B$ compare the pair
    from $A_s(j)$ with the pair from $A_u(j)$, and return the larger value
    pair for that spending level, and Stop.
```

This algorithm returns a list of undominated pairs for component Q in which the root node in Q may burn or not (since a particular pair in the final list returned may come from either $A_s(|Q|)$ or $A_u(|Q|)$. For the solution on the original tree we just consecutively merge 2 component lists at a time, deleting dominated pairs (and pairs with spending exceeding B). Once all component lists have been merged, the largest value pair that spends at most B is chosen for the full solution.

Now we prove that this algorithm produces the optimal solution. Our proof will argue about the $A_s(j)$, and $A_u(j)$ using strong induction on the ordered set of nodes and will require an internal lemma proved by induction about the $g_s(j, i)$, and $g_u(j, i)$.

We prove that the algorithm described correctly computes the optimal value.

Proof: Since the $|Q|$th node is the root, showing that $A_s(|Q|)$, and $A_u(|Q|)$ contain all non-dominated pairs corresponding to sets of edges in which every edge is strictly below the root is sufficient. We will prove this claim by strong induction on the ordered set of nodes (as ordered by the preprocessing step).

For the base case: by the form of the preprocessing step, the first node considered is a leaf in the tree Q. The leaf is either an ignition node or not, and either way the only two possibilities are those assigned by the algorithm.

The strong induction hypothesis is: suppose that for all nodes between the first and the $(j-1)$st $A_s(x)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is strictly below the parent edge of x and in which x is safe, and $A_u(x)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is strictly below the parent edge of x, and in which x is unsafe.

We will show that the same statement holds for $A_s(j)$ and $A_u(j)$. First, if j is a leaf in T then the claim obviously holds.

If j is not a leaf then by our observation and our definitions it is sufficient to show that the following lemma holds:

Lemma 2.8.1. *For each i in $\{1, 2, ...|\chi(j)|\}$, $g_s(j,i)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is weakly below the parent edge of one of the first ith children of node j and j is safe, and $g_u(j,i)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is weakly below the parent edge of one of the first ith children of node j and j is unsafe.*

Proof of Lemma: We will prove this lemma by induction. The base case is for $g_s(j,1)$ and $g_u(j,1)$. By the form of the preprocessing ordering step: the children of j are addressed by the algorithm before j is, so by the strong induction hypothesis in our proof of the theorem, $A_s(y_1)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is strictly below the parent edge of y_1 and y_1 is safe, and $A_u(y_1)$ contains all non-dominated pairs corresponding to sets in which every edge in the set is strictly below the parent edge of y_1 and y_1 is unsafe.

For $g_s(j,1)$, j is safe and either y_1 is safe or not: if y_1 is safe then only pairs that exclude the edge between them are possibly not dominated (and the algorithm includes these), if y_1 is not safe then the edge between y_1 and j must be purchased by any solution (and the algorithm includes all solutions that purchase this edge that are not dominated on account of a better pair existing below y_i with y_i unsafe-this is ensured by adding the parent edge of y_i to every pair in $A_u(y_i)$).

For $g_u(j,1)$, j is unsafe and either y_1 is safe or not: if y_1 is unsafe then only pairs that exclude the edge between them are possibly not dominated (and the algorithm includes these), if y_1 is safe then the edge between y_1 and j must be purchased by any solution (and the alg includes all solutions that purchase this edge that are not dominated on account of a better pair existing below y_i with y_i safe-this is ensured by adding the parent edge of y_i to every pair in $A_u(y_i)$). This is exactly what we need for the base case.

Now, suppose that the lemma holds for $g_s(j,i-1)$, $g_u(j,i-1)$. We want to show that it holds for $g_s(j,i)$, $g_u(j,i)$. This will involve analyzing a number of cases.

First, suppose that j is safe. Consider any set of edges R that has the property that all its edges are weakly below one of the first ith children of j and which is compatible with j being safe. Let (t, w) be the pair corresponding to R. We will show that some pair in $g_s(j, i)$ dominates (t, w). First, suppose that none of the edges in R are weakly below the ith child of j. Then, by the induction hypothesis, (t, w) is dominated by some pair in $g_s(j, i-1)$. If this is not the case then some edges of R must be weakly below the ith child of j, call the set of these edges R'.

If y_i is safe: only pairs that exclude the edge (j, y_i) are possibly not dominated. Consider the pair corresponding to the set $R \setminus R'$: by the induction hypothesis, it must be dominated by some pair in $g_s(j, i-1)$, call that pair (r, u). Consider the pair corresponding to the set R': since R' does not contain (j, y_i) (unless it is dominated, in which case we're done), by the induction hypothesis, this pair must be dominated by some pair in $A_s(y_i)$, call it (r', u'). Then the pair $(r + r', u + u')$ dominates (t, w) and is added by the algorithm to the $g_s(j, i)$ list. Transitivity of domination ensures that deletion of dominated pairs from $g_s(j, i)$ retains a pair which dominates (t, w).

If y_i is unsafe: all solutions must buy the edge (j, y_i). Consider the pair corresponding to the set $R \setminus R'$: by the induction hypothesis, it must be dominated by some pair in $g_s(j, i-1)$, call that pair (r, u). Consider the pair corresponding to the set R': since R' contains (j, y_i), by the induction hypothesis the pair corresponding to $R' \backslash (j, y_i)$ must be dominated by some pair in $A_u(y_i)$, call it (r', u'). Then the pair $(r + r', u + u' + c(e_{(y_i)})$ dominates (t, w) and is added by the algorithm to the $g_s(j, i)$ list. Transitivity of domination ensures that deletion of dominated pairs from$g_s(j, i)$ retains a pair which dominates (t, w).

Now suppose that j is unsafe. Consider any set of edges R that has the property that all its edges are weakly below one of the first ith children of j and which is compatible with j being unsafe. Let (t, w) be the pair corresponding to R. We will show that some pair in $g_u(j, i)$ dominates (t, w). First, suppose that none of the edges in R are weakly below the ith child of j. Then, by the induction hypothesis, (t, w) is dominated by some pair in $g_u(j, i - 1)$. If this is not the case then some edges of R must be weakly below the ith child of j, call the set of these edges R'.

If y_i is unsafe: only pairs that exclude the edge (j, y_i) are possibly not dominated. Consider the pair corresponding to the set $R \setminus R'$: by the induction hypothesis, it must be dominated by some pair in $g_u(j, i-1)$, call that pair (r, u). Consider the pair corresponding to the set R': since R' does not contain (j, y_i) (unless it is dominated, in which case we're done), by the induction hypothesis, this pair must be dominated by some pair in $A_u(y_i)$, call it (r', u'). Then the pair $(r + r', u + u')$ dominates (t, w) and is added by the algorithm to the $g_u(j, i)$ list. Transitivity of domination ensures that deletion of dominated pairs from $g_u(j, i)$ retains a pair which dominates (t, w).

If y_i is safe: all solutions must buy the edge (j, y_i). Consider the pair corresponding to the set $R \setminus R'$: by the induction hypothesis, it must be dominated by some pair in $g_u(j, i - 1)$, call that pair (r, u). Consider the pair corresponding to the set R': since R' contains (j, y_i), by the induction hypothesis the pair corresponding to $R' \backslash (j, y_i)$ must be dominated by some pair in $A_s(y_i)$, call it (r', u'). Then the pair $(r + r', u + u' + c(e_{(y_i)}))$ dominates (t, w) and is added by the algorithm to the $g_s(j, i)$ list. Transitivity of domination ensures that deletion of dominated pairs from $g_u(j,i)$ retains a pair which dominates (t,w). This concludes the proof of the lemma.□

Since we have proved the lemma, the definitions of $A_s(i)$, $A_s(i)$, $g_s(j,|\chi(j)|)$, and $g_u(j,|\chi(j)|)$ immediately give that the lists $A_s(j)$ and $A_u(j)$ constructed by the algorithm contain all non-dominated pairs according to their definitions.

Thus, when we compare the largest value pair at each spending level of at most B between $A_s(j)$ and $A_u(j)$ we find the maximum amount of value that can be protected by that spending level in the component Q (where the root may burn or not), as desired.

Now we convert from a pseudo-polynomial time exact algorithm to a FPTAS via rounding.

We analyze the running time of the algorithm as stated. If node j is a leaf computing $A_s(j)$ and $A_u(j)$ takes constant time. If j is not a leaf then in the worst case it has $O(n)$ children. Consider computing $g_s(j,i)$, $g_u(j,i)$ for arbitrary i in $\{1,2,...,\chi(j)\}$: as many as $O((\min(B+1,nv^*+1))^2)$ pairs are produced which then must be sorted so that the dominated pairs can be eliminated. Thus, for each node j, computing $A(j)$ takes at most $O(n(\min(B+1,nv^*+1))^2\log((\min(B+1,nv^*+1))^2)$ time, so in total the algorithm runs in $O(n^2(\min(B+1,nv^*+1))^2\log((\min(B+1,nv^*+1))^2)$. The running time of the algorithm as described is polynomial only if the numerical values are given in unary. If we had that $\min(B+1,nv^*+1)$ was polynomial in the graph size we'd have the FPTAS we want.

So, suppose that we round each of the node values down to the largest multiple of μ for some μ we will specify in a moment. The value protected by a set of edges will be at most $n\mu$ less under the rounded values than it is under the true values. If we want this difference to be no more than ϵOPT then observing that $v^* \leq OPT \leq nv^*$ (based on the v^* we specified as the maximum value of a single node that can be separated from every ignition node at cost at most B) we can set $n\mu = \epsilon v^*$ so that $\mu = \epsilon v^*/n$. Now, suppose we apply that rounding scheme and solve the problem optimally with respect to the rounded values, $v'(x)$). Letting S denote the set of nodes our solution saves and letting O denote the set of nodes that OPT saves:

$$\begin{aligned}
\sum_{x \in S} v(x) &\geq \mu \sum_{x \in S} v'(x) \\
&\geq \mu \sum_{x \in O} v'(x) \\
&\geq \sum_{x \in O} v(x) - |O|\mu \\
&\geq \sum_{x \in O} v(x) - n\mu \\
&\geq \sum_{x \in O} v(x) - \epsilon v^* \\
&\geq \sum_{x \in O} v(x) - \epsilon OPT = (1-\epsilon)OPT.
\end{aligned}$$

Thus, the solution produced is a $(1-\epsilon)$-approximation. Notice that thanks to the rounding we have that $(v^*)' = \lfloor v^*/(\epsilon v^*/n) \rfloor = \lfloor n/\epsilon \rfloor$, such that $\min(B+1, n(v^*)'+1) = O(n^2(\frac{1}{\epsilon}))$, giving total running time of $O(n^6(\frac{1}{\epsilon})^2 \log(n^4(\frac{1}{\epsilon})^2)$. Thus we have an FPTAS. □

2.8.2 Proof of Theorem 6

Theorem 6. *For a restricted class of information revelation hierarchies, there exists a bicriteria $(1-(1-\frac{1}{k\delta})^{k\delta}),\ 2+\epsilon)$-approximation algorithm for the k-stage stochastic GPP in trees provided that each ignition set has size 1 (k is a constant, δ denotes the diameter of the tree).*

Proof. For the two-stage stochastic graph protection problem in trees we explain a LP-rounding-based $(1-(1-\frac{1}{2\delta})^{2\delta})$-approximation with at most $(2+\epsilon)$ spending) which is much simpler to analyze (and only slightly worse in budget guarantee). This algorithm will be the basis of our generalization to the case of a constant number of stages.

The algorithm has an initial stage in which it *guesses* the fraction of the budget allocated to first stage spending, α. It modifies the LP by adding the constraints that every first stage decision variable with cost greater than αB is set to 0, and that every second stage decision variable with second-stage cost that is greater than $(1-\alpha)B(1+\epsilon/2)$ is set to 0. Also it adds constraints that total first stage spending is at most αB, and that total second stage spending in each scenario is at most $(1-\alpha)B(1+\epsilon/2)$. Then the algorithm solves the LP, performs pipage rounding within the first stage and within the second stage until at most one fractional value remains among the first stage decision variables and at most one fractional value remains among the second stage decisions for each scenario, and finally rounds up all of these remaining fractional values.

If the algorithm's guess of the first stage budget is bigger, but is less than $B\epsilon/2$ bigger, than what the optimal solution spends in the first stage, then $(1-\alpha)B(1+\epsilon/2)$ is larger than what the optimal solution spends in the second stage and the new constraints that were added to the LP do not affect the feasibility of the optimal solution. Pipage rounding terminating in rounding up all final fractional values gives an upperbound on a $(1-(1-1/2\delta)^{2\delta})$ fraction of the modified LP value (where d is is the tree diameter), so since the modified LP has the optimal value as a feasible solution the algorithm gives a $(1-(1-1/2\delta)^{2\delta})$-approximation for the optimal amount of expected protected value.

Further, the cost of every solution found in this way (regardless of whether the algorithm guessed right about the budget division) is at most $2\alpha B + 2(1-\alpha)B(1+\epsilon/2) \leq B(2+\epsilon)$ (since the solution cost does not change in pipage steps, and the additional constraints that we added to the LP bound the increase when the final fractional variables are rounded up to 1).

If the algorithm makes $2/\epsilon$ guesses for α of $\{0, \epsilon/2, \epsilon, 3\epsilon/2, 2\epsilon, ..., 1\}$, some guess will be at most $B\epsilon/2$ bigger than what the optimal solution spends in the first stage, and thus will give a $(1-(1-1/2\delta)^{2\delta})$-approximation for the optimal expected value protected.

So, the algorithm can solve $2/\epsilon$ LPs, round them via pipage, and round up the final values to obtain a solution which is $(1-(1-1/2\delta)^{2\delta})$-value protecting and $(2+\epsilon)$-budget approximate.

2.8.3 Proof of k-stage Stochastic Graph Protection and k-stage MCKP.

We can use our algorithm of guessing the spending division of the optimal solution, modifying the natural LP, solving, applying pipage rounding, and rounding up the remaining fractional values for the case of a constant number of stages if we restrict the form of the updates.

The k-stage stochastic graph protection problem in trees (for constant k) has k stages in which information is revealed and decisions about edge removal are made (rather than one or two stages). This information can be considered as updates that arrive at k specific times which condition the distribution on where the ignition will occur (by specifying that the ignition will occur among some particular subset of the nodes).

The input includes a set of partitions of the node set: in the first stage the partition is simply the whole node set (and there is a single set of first-stage costs for removing edges from the graph), the second-stage partition refines the first-stage partition (and for each second-stage partition piece there is a specific set of second-stage costs for edge removal), the third-stage partition refines the second-stage partition (and for each third-stage partition piece there is a specific set of third-stage costs for edge removal), etc. The final stage has a partition in which each node is in its own partition piece (that is, the ignition point is realized and there is a final chance to remove edges at a scenario-specific cost). In each stage the planner has the option to remove additional edges from the graph at some (stage, partition piece)-specific cost. That is, a solution specifies which edges will be removed for each partition piece at each stage. The total cost incurred for each sequence of k partition pieces (where one partition piece is chosen from each stage and later stage pieces refine earlier stage pieces) should be B.

Our analysis showing a $(1-(1-1/2\delta)^{2\delta})$-value protecting and $(2+\epsilon)$-budget approximate solution will extend to this case when there are at most a polynomial number of possible guesses for the budget division (since each such guess requires that we solve a modified LP, pipage round and then round up remaining fractional variables): in general this will be the case when the sum of the number of partition pieces over all stages excepting the last stage (in which there are n partition pieces) is bounded by a constant χ. In particular, there are at most

$$\Big(\frac{(1+\epsilon)k}{\epsilon}\Big)^{\chi}$$

possible right-hand-sides of constraints for the extra budget constraints (corresponding to budget divisions) of the modified LP since the necessary coarseness for rounding up all fractional values post-pipage to incur a loss of at most ϵ is ϵ/k, and a budget division must specify χ values (one per partition piece in each stage excepting the last stage). This later fact can be observed by inspection of 3-stage and 4-stage problems.

Now, decision variables will specify whether or not to buy each edge in each stage depending on which partition piece of the stage is realized. We can think of the final realized ignition as being *at the bottom* of a sequence of partition pieces in which each is from a different stage and the later stage partition pieces are subsets of the earlier stage partition pieces. We will let y_e^{jq} denote the decision variable corresponding to the edge e, the stage j, and the qth partition piece from stage j.

The value of a node v is protected in a scenario i when there is a (stage, partition piece, edge) pair where i is in the partition piece and the edge is on the path in T from i to v whose corresponding decision variable is set to 1. Suppose that the ignition node for scenario i is at the bottom of partition sequence $q_1, q_2, ..., q_k$. Then we have a constraint of form:

$$\underbrace{\sum_{e \in P(i,v)} y_e^{1q_1}}_{\text{protect } v \text{ from } i \text{ in 1st stage}} + \underbrace{\sum_{e \in P(i,v)} y_e^{2q_2}}_{\text{protect } v \text{ from } i \text{ in 2nd stage}} +...$$

$$... + \underbrace{\sum_{e \in P(i,v)} y_e^{kq_k}}_{\text{protect } v \text{ from } i \text{ in } k\text{th stage}} \geq x_{iv} \quad \forall (i,v).$$

Similarly, our original budget constraint for scenario i is

$$\underbrace{\sum_e y_e c_e}_{\text{1st stage spending}} + \underbrace{\sum_e y_e^{2q_2}(M_e^{q_2} c_e)}_{\text{2nd stage spending}} +... + \underbrace{\sum_e y_e^{2q_k}(M_e^{q_k} c_e)}_{k\text{th stage spending}} \leq B \quad \forall i,$$

where $M_e^{q_j}$ is the inflation factor associated with edge e by partition piece q_j (implicitly this is stage-specific since it is partition-piece specific).

So, using $q_1(i), q_2(i), ..., q_k(i)$ to denote the partition sequence for the ignition node of scenario i, we get:

$$\max \sum_{(i,v)} (p_i v_v) x_{iv}$$

$$s.t.$$

$$\sum_{e \in P(i,v)} y_e^{1q_1(i)} + \sum_{e \in P(i,v)} y_e^{2q_2(i)} + ... + \sum_{e \in P(i,v)} y_e^{kq_k(i)} \geq x_{iv} \quad \forall (i,v)$$

$$\sum_e y_e c_e + \sum_e y_e^{2q_2(i)} (M_e^{q_2(i)} c_e) + \sum_e y_e^{3q_3(i)} (M_e^{q_3(i)} c_e) + ...$$

$$... + \sum_e y_e^{2q_k(i)} (M_e^{q_k(i)} c_e) \leq B \quad \forall i$$

$$x_{iv} \leq 1 \quad \forall (i,v)$$

This is the analog of our original LP, and now for each guess of the budget division we impose stage specific budget constraints, and also set to zero decision variables that correspond to single-edge purchases which would violate the budget division limit on in-stage spending (notice that this depends on the $M_e^{q_j}$).

Now one of the guesses of budget division made by the algorithm will have that the guess for each stage is larger than what the optimal spends for that stage, but is at most $B\epsilon/k$ larger than what the optimal spends for that stage: the optimal solution will be feasible for the modified LP so that the solution post-pipage rounding the LP optimal and rounding up the final fractional variables will be a $(1-(1-1/k\delta)^{k\delta})$-approximation of expected value that can be saved ($k\delta$ is the largest possible number of edge decision variables in constraints of the first form, so the property 2 for F has k in the place of 2 now). Rounding in the pipage stage is now over pairs of fractional variables for a common (stage, partition piece) pair (this is the natural analog of the *for a common i* condition that we imposed for pairs of second stage variables in the 2-stage version). Thus, the choice of the most expected value-protecting solution over those produced by all budget division guesses will be a $(1-(1-1/k\delta)^{k\delta})$-approximation of expected value that can be saved.

Also, we again have that disqualifying decision variables that would incur purchase costs greater than the budget division limit for that stage bounds the effect of rounding up the final remaining fractional variables so that every solution found is a $(2+\epsilon)$-budget approximate solution. □

2.8.4 Proof of Theorem 7.

Theorem 7. *There exists a $(1-\frac{1}{e})$-approximation algorithm for the single-stage stochastic GPP in trees with probabilistic edge transmission (provided that each ignition set has size 1). For MCKP with probabilistic element failure: there exists a $(1-\frac{1}{e})$-approximation algorithm.*

Proof (MKCP case): The analysis is basically identical to that for the stochastic GPP with probabilistic edge transmission. We generalize MCKP by introducing the feature that we have the option to partially buy an item: a partially bought item can still probabilistically fail to materialize (and thus fail to cover the sets that contain it). The precise statement is below.

We'll call the elements in the MCKP problem *items*. The maximum coverage problem with a knapsack constraint with probabilistic item failure is: given a family $F = \{S_j : j \in J\}$ of subsets with weights w_j of a set $I = \{1, 2, ..., n\}$ of items where each item has an associated step function that specifies probability of failure as a function of spending level and a cost c_j, and positive integer B, choose a spending level for each item with total spending at most B so as to maximize the expected total weight of the sets in F having nontrivial intersections with the realized non-failing set of items. We say a set having non-trivial intersection with the realized set of non-failing items is *covered* by the realized set, so we aim to maximize the expected weight covered. The expectation is over realizations of which items fail, where each item fails with independent probability that depends according to the step function on the spending level we chose for that item. We require that the steps of each step function be specified explicitly in the input (so that there are only polynomially many steps for each item).

Notice that this probabilistic model generalizes the previous MCKP model since in the previous model we had the option to either pay 100% of the item cost to reduce the probability of failure to 0, or the option to pay 0% of the item cost to leave the probability of failure at 1.

We consider each (spending level, item) pair an element our solution can buy with cost corresponding to the spending level times the item cost (we only have elements corresponding to critical spending levels at which the failure probability instantaneously drops). Denote the total set of such elements by X. The expected weight covered is a set function over these elements (where expectation is over a realization of item failure for each item). Denote this function by

E. We wish to maximize this set function by buying elements subject to a knapsack constraint. If we are able to show that this set function is submodular, [48] will immediately yield a $(1 - \frac{1}{e})$-approximation that is budget-balanced (provided that we can compute in polynomial time the improvement of E when a (spending level, item) pair is added to an existing set).

To prove submodularity we will establish that the law of diminishing returns holds: for an arbitrary (spending level, item) pair denoted by a, if $A \subseteq B \subseteq X$, then

$$E(A \cup a) - E(A) \geq E(B \cup a) - E(B).$$

For this we will need some notation. Let the item of the (spending level, item) pair a be denoted by e.

The item e has a step function with levels of spending and the resulting decreasing sequence of failure probabilities over e. Let $\alpha_1, \alpha_2, ...$ denote the decreasing sequence of failure probabilities that may be achieved by the ordered set of increasing spending levels (α_1 is achieved with no spending). So a results in some probability of failure α_i. Before a is added, A contains some set of elements which affect the failure probability of e, and B contains a superset of those elements. Thus the probability of of failure of item e is (weakly) larger for the set A than for the set B. When a is added to each set, in both cases the new probability of failure for item e is the minimum of α_i and the current probability of failure for item e. The gap when a is added is larger for A than for B. Letting $\wp_e(\cdot)$ denote the probability of failure for item e as a function of the set of elements:

$$\wp_e(A) \geq \wp_e(B) \Rightarrow \wp_e(A) - \wp_e(A \cup a) \geq \wp_e(B) - \wp_e(B \cup a)$$

Next, focus on a particular subset S_j from our family. Either S_j does not contain e (in which case the addition of e leaves the expected coverage of S_j unchanged) or it does. For the later case, the probability of every one of the other items in S_j failing (excepting e) is at least for A what it is for B. Let $P(A)$ denote the probability that every one of the items in S_j fails under A, and let $P(B)$denote the probability that every one of the items in S_j fails under B. Then

$$\begin{aligned}
P(A) \geq P(B) &\Rightarrow P(A)(\wp_e(A) - \wp_e(A \cup a)) \geq P(B)(\wp_e(B) - \wp_e(B \cup a)) \\
&\Rightarrow P(A)\wp_e(A) - P(A)\wp_e(A \cup a) \geq P(B)\wp_e(B) - P(B)\wp_e(B \cup a) \\
&\Rightarrow (1 - P(A)\wp_e(A \cup a)) - (1 - P(A)\wp_e(A)) \geq \\
&\qquad\qquad (1 - P(B)\wp_e(B \cup a)) - (1 - P(B)\wp_e(B)) \\
&\Rightarrow E(A \cup a) - E(A) \geq E(B \cup a) - E(B).
\end{aligned}$$

The second to last line compares the changes in the probability that the set S_j is covered which result when a is added to A and when a is added to B. The final inequality follows from simply summing the change in expected covered weight over all $S_j \in F$ (including those for which the addition of e caused no changes in the probability of coverage). This establishes submodularity.

Additionally, the time to compute the change in E resulting from the addition of a single element simply requires computing the product of failure for all items in S_j twice for each S_j. This takes polynomial time in the input size. Then, we are required to do this for each possible element (there are polynomially many elements). Thus, [48] gives a $(1 - \frac{1}{e})$-approximation for our problem.

Our approximation guarantee for this stochastic generalization exactly matches the asymptotic guarantee obtained in [1] for the special case in which elements cover all sets which contain them with probability 1 if we 100% buy the element, and with probability 0 if we 0% buy it.

2.8.5 Reduction of wBSMC in trees with probabilistic-edge occurrence to MCKP with probabilistic element failure.

The probabilistic edge occurrence generalization of wBSMC in trees immediately reduces to the probabilistic item failure generalization of MCKP:

The set I of ground items contains two kinds of items. The first kind of item corresponds to (node, node)-pairs. Each such item has cost 2B. The probability of failure for this type of item is 1 regardless of spending level.

The second kind of items correspond to individual edges. Each such item has cost c_e according to the edge it corresponds to. The probability-of-failure step function for this item is just the probability-of-occurrence step function for the corresponding edge. For each (node $= v$, node$= u$)-pair, the family F of subsets has a subset which contains the (v, u)-pair as well as the items corresponding to the edges on the path in T between v and u. This subset has weight equal to the weight awarded if v and u are separated in the wBSMC problem. This is the family of all subsets given nonzero weight.

For wBSMC, suppose that Y is a set of spending levels of cost at most B. Then the set of matching spending levels X on items of I corresponding to edges in Y costs the same amount. Further, Y separates v and u with probability equal to 1 minus the probability that every edge on the path from v to u occurs: equivalently the set containing the pair (v, u) is covered with probability equal to 1 minus the probability that every item in the set fails. Thus, expected weight separated by Y is exactly equal to the expected weight of subsets intersected by X. For any budget-balanced solution, the budget spent is the same in both cases and the expected weight separated in wBSMC is exactly the expected weight of subsets intersected in the MCKP instance described.

Our approximation guarantee for this stochastic-edge-occurrence generalization in trees exactly matches the guarantee obtained in [15] for the deterministic case in trees.

CHAPTER 3

ROBUST CUTS OVER TIME: COMBATTING THE SPREAD OF INVASIVE SPECIES WITH UNRELIABLE BIOLOGICAL CONTROL

A version of the work in this chapter will appear in the Proceedings of the 26th Conference on Artificial Intelligence (AAAI 2012) [47].

3.1 Summary

Widespread accounts of the harmful effects of invasive species have stimulated both practical and theoretical studies on how the spread of these destructive agents can be contained. In practice, a widely used method is the deployment of *biological control agents*, that is, the release of an additional species (which may also spread) that creates a hostile environment for the invader. Seeding colonies of these protective biological control agents can be used to build a kind of living barrier against the spread of the harmful invader, but the ecological literature documents that attempts to establish colonies of biological control agents often fail (opening gaps in the barrier). Further, the supply of the protective species is limited, and the full supply may not be available immediately. This problem has a natural temporal component: biological control is deployed as the extent of the harmful invasion grows. How can a limited supply of unreliable biological control agents best be deployed over time to protect the landscape against the spread of a harmful invasive species?

To explore this question we introduce a new family of stochastic graph vaccination problems that generalizes ideas from social networks and multistage

graph vaccination. We point out a $(1 - 1/e)$-approximation algorithm for a deterministic base case studied in the social networks literature (matching the previous best randomized $(1 - 1/e)$ guarantee for that problem). Next, we show that the randomized $(1 - 1/e)$ guarantee (and a deterministic 1/2 guarantee) can be extended to our much more general family of stochastic graph vaccination problems in which vaccinations (a.k.a. biological control colonies) spread but may be unreliable. For the non-spreading vaccination case with unreliable vaccines, we give matching results in trees. Qualitatively, our extension is from computing "cuts over time" to computing "robust cuts over time."

Our new family of problems captures the key tensions we identify for containing invasive species spread with unreliable biological control agents: a robust barrier is built over time with unreliable resources to contain an expanding invasion.

3.2 Introduction

The devastating impacts of *invasive* (a.k.a. non-native) species are being increasingly documented around the globe. Both plant and animal invaders can spread across landscapes, quickly overwhelming local ecosystems through direct competition for resources and predation. In addition to ecological concerns, economic forecasts predict massive ramifications on renewable natural resource industries (e.g. timber) and residential land values [32]. These accounts have stimulated both practical and theoretical studies on how the spatial spread of these destructive agents can be contained [33] [44].

A commonly-practiced technique for invasive species containment is the introduction of a *biological control agent*, that is, another species (which may also spread) whose presence renders the habitat inhospitable to the harmful invasive species. A problem explored in the ecological literature is that introductions of these control species often fail to establish viable colonies [43]. The probability of establishment is often related to the size of the seed colony [43], but since supplies of the biological control agent are usually limited, and breeding additional agents takes time, land managers must proceed with unreliable deployments. In addition to the many unresolved scientific and ecological questions about biological control strategies (particulary as regards unintended negative consequences of control species introductions), there are basic mathematical issues here that are not well understood. Where should a limited supply of unreliable biological control agents be deployed over time to contain the spread of a harmful invader?

To explore this question we introduce a new family of stochastic graph vaccination problems that generalizes ideas from social networks and multistage graph vaccination.

Literature Review. Hartnell's Firefighter Problem [26] concerns the containment of an infection outbreak in a graph when the amount of action per time step is limited. In each time step all neighbors of an infected node become infected. The planner has the option to vaccinate k nodes per time step (so that they will never become infected). Where should these vaccinations be placed at each time-step to minimize the total number of nodes infected? Qualitatively, the goal is to compute the best "cut over time."

In addition to literal diseases that can be spread between contacts in social networks, recent work on disease spread and vaccination in networks has proposed the notion of harmful ideas or ideology spreading through social networks. The antidote to harmful ideology is positive ideology or ideas which also spread through the network vaccinating people who are exposed to them against the harmful ideology [3]. Once an individual is convinced by either harmful or positive ideology we suppose that they can no longer be converted to the opposing viewpoint. Considering this setting with an additional temporal component has motivated interest in a spreading-vaccination version of the Firefighter problem.

For the spreading-vaccination version of the Firefighter problem (in which vaccination also spreads to neighbors at the same rate as infection) Anshelevich, Chakrabarty, Hate, and Swamy show that the problem reduces to maximizing a submodular function subject to a partition-matroid constraint [3]. They use this reduction in conjunction with the recent result of Calinescu, Chekuri, Pál and Vondrák [11] for maximizing a submodular function subject to a matroid-constraint to give a randomized $(1 - 1/e)$-approximation (or to get a deterministic $1/2$ approximation by applying a greedy result of Fisher, Nemhauser and Wolsey [19]).

In fact, we note that since [3] show that the spreading-vaccination Firefighter objective is a coverage function, a deterministic $(1 - 1/e)$-approximation can be obtained immediately by reducing the problem to maximum coverage subject to group-constraints and applying a decade-old result of Ageev and Sviridenko for that problem [1]. This deterministic $(1-1/e)$ guarantee is tight (unless $NP \subseteq DTIME(n^{\mathtt{polylog}(n)})$ due to another result from [3]), and it matches the current best guarantee in trees for the non-spreading Firefighter Problem due to Cai, Verbin, and Yang [10].

In the ecology literature, Shea and Possingham give curves that describe probability of control colony survival as a function of initial population [43], and point out that biological control resources may not all be available at the start of a suppression effort. Based on field experiments, Norris, Memmott and Lovell tune parameters impacting colony survival of an insect control, thrips (*Sericothrips staphylinus*), for the models in [43]; they find that the model from [43] explains a very large fraction of the variation in the data and that parameters depend significantly on stochastic weather events (rainfall) [38].

As mentioned in [44], tree landscapes are used to study invasive-species spread through stream and river systems (for example, [14] explores how different stream branching affects invasions). Invasions of tree-landscape topologies also occur in the riparian-vegetation zones bordering river systems. For example, Tamarisk is an exotic noxious weed that invades such zones, driving out local wetland plants; the Colorado Department of Agriculture has suppressed Tamarisk invasion using the tamarisk leaf beetle, *Diorhabda elongata* along the Dolores, Colorado, Yampa and Green Rivers [13].

Our Results. The conditions required to apply the result of Calinescu, Chekuri, Pal and Vondrák (CCPV) are more general than are required for the spreading-vaccination Firefighter Problem: in exploring how much can we generalize the spreading-vaccination Firefighter Problem and still benefit from reducing to maximizing a submodular function subject to a partition-matroid constraint, we introduce a natural new family of general stochastic cut problems that capture the key tensions in containing invasive species with unreliable biological control agents. Applying CCPV and [19]), we immediately obtain randomized $(1-1/e)$ and deterministic 1/2-approximations for the spreading cases in graphs and for the non-spreading cases in trees).

Our Generalization: Unreliability. At the conclusion of their paper, Anshelevich, Chakrabarty, Hate, and Swamy [3] raise the very natural question of what happens when vaccination is less virulent than infection. In particular, they mention why a positive spread rate for vaccination of less than 1 voids a reduction similar to the rate 1 case. We introduce an alternative way of blunting the strength of vaccination: spread of vaccination remains deterministic at rate 1 but we allow the possibility that a vaccine will be faulty with some probability (which is given in the input). First we give a direct generalization of the Firefighter Problem (in which k interchangeable and probabilistically-successful vaccines are distributed per time step), then we generalize further.

One interesting model among our new class considers vaccines that become better with time (their failure probability decreases) but which may be deployed at most once over the time horizon; what is the correct balance between acting early with unreliable resources when the extent of the infection is small and acting later with more reliable resources once the infection is more extensive? This corresponds to the tradeoff between immediately deploying small biological control colonies (which have high failure probability) vs. postponing deployment while the invasive species spreads in order to breed larger and more reliable biological control colonies under protected conditions.

We describe our models as offline problems: all decisions are made without knowing which vaccines will be realized as effective. In the online setting where reliability/unreliability is realized and observed at each time step, simply extracting nodes which will be covered by a vaccination that has already been realized as effective (which is polynomial-time computable) will give a new problem of an identical type (so that our approximation guarantees hold from that time step forward).

Qualitatively, our new notion could be described as computing "robust cuts over time" in which unreliable vaccine resources are deployed over a series of time steps in order to cut (or cover, in the spreading case) a graph against the spread of an infection. In terms of idea-spread through social networks we could describe this as a problem of waging an ideological war when we are unsure which positive ideologies we plant with "go viral." An interesting feature of this expanded model is that the optimal solution may repeatedly target nodes which are in some sense "highly connective." In these features our results add to the exploration begun in the seminal paper of Kempe, Kleinberg, and Tardos on maximizing influence spread in social networks [31] which was motivated by emerging applications in viral marketing. Our results for the non-spreading cases in trees are closely related to the recent work of Shmoys and Spencer [44] on containing stochastic outbreaks of invasive species in tree landscapes via imperfect (a.k.a. probabilistic) edge removal; that work addresses only connectivity vs. disconnectivity and lacks the natural temporal nature of an expanding invasion that we address here.

We show that for the cases of spreading-vaccination in graphs and non-spreading vaccination in trees the general stochastic models we describe reduce to maximizing a submodular objective subject to a partition-matroid constraint. Thus, the result of Calinescu, Chekuri, Pal and Vondrák (CCPV) [11] gives a randomized $(1 - 1/e)$-approximation (or we can get a deterministic $1/2$ approximation by applying an older result of Fisher, Nemhauser and Wolsey [19]). We will need that the failure of vaccines are independent events so that we can compute the values required by the CCPV method in polynomial time.

3.3 From Classical Firefighter Models to Stochastic Graph Vaccination

We are given a directed graph $G = (V, E)$ and a source node s. Denote $|V|$ by n, and $|E|$ by m. All nodes in the graph can have one of three states: *infected*, *vaccinated*, or *vulnerable*. At time $\tau = 0$, node s is infected and all other nodes are *vulnerable*. Once a node has become *infected* or *vaccinated* its state will never change. The goal of the planner is to maximize the expected number of members of the network that are protected from infection.

- **Classical (non-spreading):** At time $\tau = i > 0$ at most $k \leq n$ vaccines can be deployed at vulnerable nodes, where each vaccine has probability 1 of effectiveness (and probability 0 of not being effective). When a vaccine is realized as effective, then the node it was deployed at becomes *vaccinated*. In time step $i+1$ all *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.

- **Classical (spreading-vaccination):** At time $\tau = i > 0$ at most $k \leq n$ vaccines can be deployed at vulnerable nodes, where each vaccine has probability 1 of effectiveness (and probability 0 of not being effective). When a vaccine is realized as effective, then the node it was planted at becomes *vaccinated*. In time step $i+1$ all *vulnerable* nodes reachable by a single edge from any *vaccinated* node become *vaccinated*. Then, in the same time step, all remaining *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.

First we give the direct stochastic generalizations of these models:

- **Stochastic (non-spreading):** At time $\tau = i > 0$ at most $k \leq n$ vaccines can be deployed at vulnerable nodes, where each vaccine has independent probability p_i of effectiveness (and probability $1-p_i$ of not being effective). When a vaccine is realized as effective, then the node it was deployed at becomes *vaccinated*. In time step $i+1$ all *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.
- **Stochastic (spreading-vaccination):** At time $\tau = i > 0$ at most $k \leq n$ vaccines can be deployed at vulnerable nodes, where each vaccine has independent probability p_i of effectiveness (and probability $1 - p_i$ of not being effective). When a vaccine is realized as effective, then the node it was planted at becomes *vaccinated*. In time step $i+1$ all *vulnerable* nodes reachable by a single edge from any *vaccinated* node become *vaccinated*.

> Then, in the same time step, all remaining *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.

While we've specified that $k \leq n$ above, in fact any k that is bounded above by a polynomial in n will work. Our choice that vaccination prevails when exposure to vaccination and infection is simultaneous is not required for any of the described guarantees.

Figure 3.1 below motivates our next generalization: we relax the idea that there is only a single timestep in which a particular vaccine is effective.

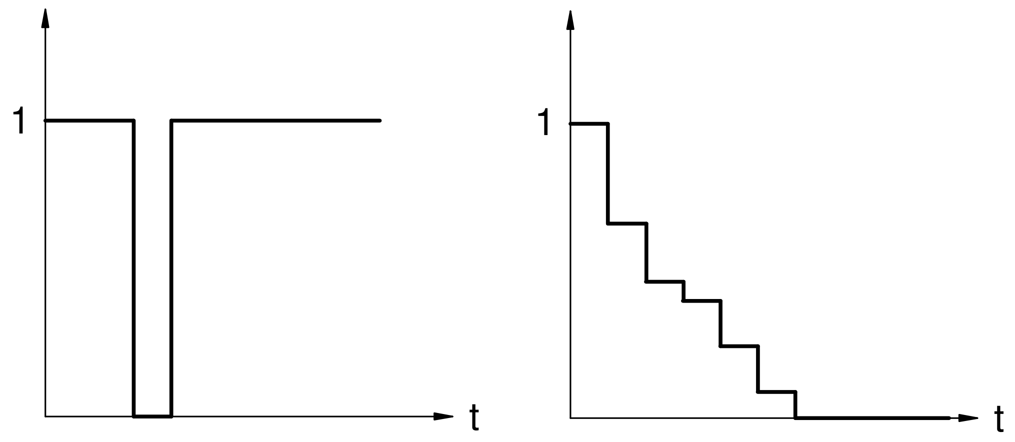

Figure 3.1: **Examples of $f_j(x,t)$:** On the left a vaccine fails with probability 1 except during timestep 3 in which it is effective with probability 1. On the right, a vaccine has a failure probability that decreases with time. Not shown here: the dependence of $f_j(x,t)$ on the node x at which j is deployed can be used to capture that some sites are more suitable for the establishment of biological control colonies.

- **General Stochastic Graph Vaccination (non-spreading):** The input gives a set of vaccines J. For each $j \in J$ the input gives a function $f_j(x, t) : \{1, 2, ..., n\} \times \{1, 2, ..., n\} \to [0, 1]$ that describes the effectiveness probability of j if it is deployed at node x at time t. Each vaccine is deployed at some node at some time step, and effectiveness of each vaccine is realized independently. When a vaccine is realized as effective, then the node it was deployed at becomes *vaccinated*. In time step $i + 1$ all *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.

- **General Stochastic Graph Vaccination (spreading-vaccination):** The input gives a set of vaccines J. For each $j \in J$ the input gives a function $f_j(x, t) : \{1, 2, ..., n\} \times \{1, 2, ..., n\} \to [0, 1]$ that describes the effectiveness probability of j if it is deployed at node x at time t. Each vaccine is deployed at some node at some time step, and effectiveness of each vaccine is realized independently. When a vaccine is realized as effective, then the node it was planted at becomes *vaccinated*. In time step $i + 1$ all *vulnerable* nodes reachable by a single edge from any *vaccinated* node become *vaccinated*. Then, in the same time step, all remaining *vulnerable* nodes reachable by a single edge from any *infected* node become *infected*.

It is immediately obvious that the classical problems are special cases of the stochastic problems (setting $p_i = 1$ for all i), but seeing that the stochastic problems are special cases of the general stochastic problems requires a slightly subtle observation.

Since the diameter of the graph is at most n, at time $\tau = n$ every node of the graph will either be infected or we will know that it will never be infected [3].

Thus, in the stochastic problem there is no need to consider any vaccine deployments at time steps beyond n. This is true for both the spreading-vaccination and non-spreading problems. Consider an instance of the stochastic problem: for each time-step i, create k copies of vaccine j which has $f_j(x, i) = p_i$ for all $x \in V$, and $f_j(x, i') = 0$ for all $i' \neq i$. This gives an input for the most general problem with kn vaccines.

Notice that the stochastic elements we introduce can cause the form of the solutions to be very different than in the deterministic vaccination case: in the typical spreading-vaccination Firefighter model once a node is vaccinated its status is known conclusively, and as each time step elapses expanding neighborhoods around it are removed from the set that could possibly be vaccinated in an optimal solution, while in our stochastic model the optimal solution may repeatedly target some "important nodes" in a single time step or repeatedly over a series of time steps. See Figure 3.2 for an example.

Further, if a larger node cut could protect a large portion of the network but cannot be reliably imposed with the vaccines available, a moderate portion of the network may be sacrificed to more reliably impose a smaller cut further from the source. The most general model softens the hard limits on how much vaccination can be performed per time step: the freedom to design $f_j(x, t)$ allows exploration of a number of interesting tensions: vaccines that each become more powerful with time but can be deployed only once, vaccines that will be most effective within a particular time step (as in the stochastic case), but can also be shifted to neighboring time steps in exchange for a penalty on their reliability, etc.

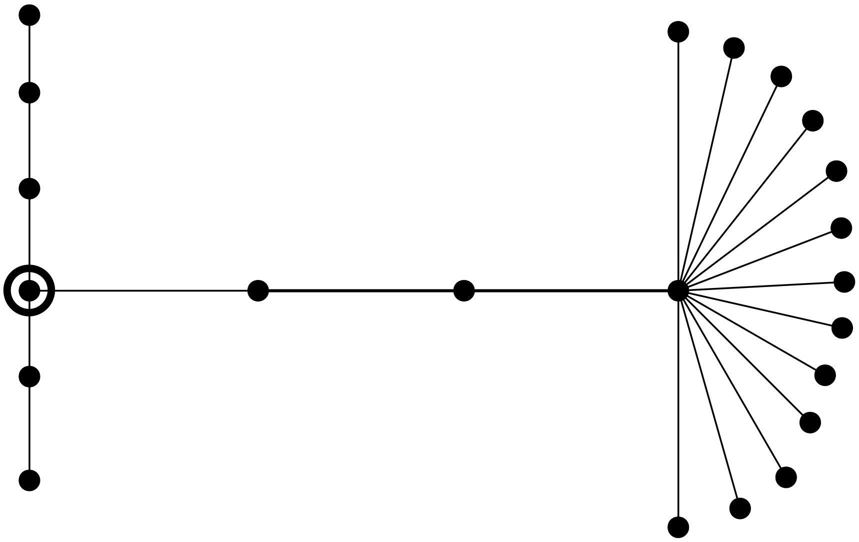

Figure 3.2: **Repeated-Targeting Example:** The initially-infected node is circled. The optimal solution to the robust-cuts-over-time problem will repeatedly target the long path leading to the star.

3.4 Hardness

The classical case of non-spreading Firefighter is NP-complete in trees. The current best guarantee in general trees is a deterministic $(1 - 1/e)$-approximation due to Cai, Verbin, and Yang [10]. Some polynomial-time algorithms exist for restricted classes of trees. We give a randomized $(1 - 1/e)$-approximation for the General Stochastic Graph Vaccination Problem (non-spreading) described above in trees.

The classical case of the spreading-vaccination Firefighter Problem is hard to approximate within $c > (1-1/e)$ unless $NP \subseteq DTIME(n^{\texttt{polylog}(n)})$ [3]. We'll give a randomized $(1-1/e)$-approximation for the General Stochastic Graph Vaccination Problem with spreading vaccination.

3.5 Reductions to Maximum Coverage Subject to Probabilistic Element Failure

Theorem 3.5.1. *The non-spreading case of the General Stochastic Graph Vaccination Problem in trees reduces in polynomial time to the problem of maximizing a (polynomial-time computable) submodular function subject to a partition-matroid constraint. Thus, applying CCPV [11] gives a randomized $(1-1/e)$-approximation algorithm, and applying [19] gives a deterministic $1/2$-approximation algorithm.*

Theorem 3.5.2. *The spreading-vaccination case of the General Stochastic Graph Vaccination Problem reduces in polynomial time to the problem of maximizing a (polynomial-time computable) submodular function subject to a partition-matroid constraint. Thus, applying CCPV [11] gives a randomized $(1-1/e)$-approximation algorithm, and applying [19] gives a deterministic $1/2$-approximation algorithm.*

In fact we will show that both problems reduce in polynomial-time to Maximum Coverage subject to probabilistic set failure (which we'll show is a submodular function, similar to the argument in [44]) subject to a partition-matroid constraint.

Maximum Coverage subject to probabilistic set failure: Let S be a collection of sets $S_1, S_2, ..., S_n$ over a ground set of elements Y. Each set S_i has an associated independent failure probability p_i. The objective is to choose a set $O \subseteq S$ such that the expected cardinality of the union of sets realized from O according to the p_i is maximized. *With a partition-matroid constraint:* For some partition of S into $P_1, ..., P_q$, at most k_j sets from P_j can be included in O.

Reduction 1: Non-spreading General Stochastic Graph Vaccination in trees (nSGVt) to Maximum Coverage subject to probabilistic set failure with partition-matroid constraint (pMCPM).

Let r denote the node where the infection starts. For each node x in T: let d denote the distance from x to r. The ground set of elements in the instance we construct is the set of nodes of tree T which we'll call Y. For every pair of $j \in J$ and $t \in \{1, 2, ...d\}$, create a set S_{jxt} which contains all nodes y whose shortest path to r contains x. Let $p_{jxt} = 1 - f_j(x, t)$. This gives at most $|J| \cdot |V| \cdot |V|$ sets which can each be computed in polynomial time. Define a partition of the collection of sets by letting all sets which have a common j be in a single partition piece, and specify that at most one set can be selected from each partition piece.

Let O denote the optimal solution of the resulting pMCPM instance, and use it to construct a solution for nSGVt as follows. If $S_{jxt} \in O$ then deploy vaccine j at node x at time t. Since O obeys the partition-matroid constraint this constructed solution deploys vaccine j at most once (at some particular node, at some particular time). What is the expected number of nodes saved from infection by the constructed strategy? Using linearity of expectation, it is the sum over all $y \in V$ of the expectation of an indicator variable that is 1 when y is saved, and 0 otherwise.

A node y is saved from infection exactly when some node on the path from r to y is *vaccinated effectively* before the infection arrives at that node; that is, when not every vaccination attempt on nodes on the r to y path between time 0 and $d(r, y)$ fails. By our construction, this event (and its probability) corresponds exactly to the event that y is in some non-failing set that contains it from O. Thus, the expected number of nodes saved from infection is exactly the expected cardinality of union of realized sets from O.

In the reverse direction, every valid solution for nSGVt defines a valid solution for the pMCPM instance: if vaccine j is to be deployed at node x at time t, include S_{jxt} in the solution. Since each vaccine is deployed at most once, the partition-matroid constraint is obeyed. Because the objective value exactly coincides, if the solution we constructed from O were not optimal for the nSGVt instance, it would imply that O was not optimal for the pMCPM instance.□

In the following reduction we add a probabilistic component to an observation about coverage of nodes by expanding vaccinated sets from [3], then exploit the fact that only $|V|$ time steps must be considered so that the partition can force a single deployment of a vaccine (since the requirement that only k vaccines are deployed per time step need no longer be explicitly articulated via a partition-matroid constraint as in [3]).

Reduction 2: Spreading-vaccination General Stochastic Graph Vaccination

(sSGV) to Maximum Coverage subject to probabilistic set failure with partition-matroid constraint (pMCPM).

Let r denote the node where the infection starts. For any two nodes x and y, let $d(x, y)$ denote the distance from x to y. The ground set of elements in the instance we construct is the set of nodes of G which we'll call Y.

For every triple (j, x, t) of $j \in J$, $x \in V$, and $t \in \{1, 2, ..., n\}$, create a set S_{jxt} which contains all nodes y for which $d(x, y) \leq d(y, r) - t$. Let $p_{jxt} = 1 - f_j(x, t)$. This gives at most $|J| \cdot |V| \cdot |V|$ sets which can each be computed in polynomial time. Define a partition of the collection of sets by letting all sets which have a common j be in a single partition piece, and specify that at most one set can be selected from each partition piece.

Let O denote the optimal solution of the resulting pMCPM instance, and use it to construct a solution for sSGV as follows. If $S_{jxt} \in O$ then deploy vaccine j at node x at time t. Since O obeys the partition-matroid constraint this constructed solution deploys vaccine j at most once (at some particular node, at some particular time). What is the expected number of nodes saved from infection by the constructed strategy? Using linearity of expectation, it is the sum over all $y \in V$ of the expectation of an indicator variable that is 1 when y is saved, and 0 otherwise.

A node y is saved from infection exactly when there exists some vaccine deployment that is realized as effective at node x and time t such that vaccination spreading from x starting at t reaches y before infection reaches y. That is, when $d(x, y) \leq d(r, y) - t$ and a vaccine deployed at x at time t is realized as effective.

By our construction, this event (and its probability) corresponds exactly to the event that y is in some non-failing set that contains it from O. Thus, the expected number of nodes saved from infection is exactly the expected cardinality of union of realized sets from O.

In the reverse direction, every valid solution for sSGV defines a valid solution for the pMCPM instance: if vaccine j is to be deployed at node x at time t, include S_{jxt} in the solution. Since each vaccine is deployed at most once, the partition-matroid constraint is obeyed. Because the objective value exactly coincides, if the solution we constructed from O were not optimal for the sSGV instance, it would imply that O was not optimal for the pMCPM instance.□

The following result is similar to that in described by analogy in Chapter 2 (and [44]) for a knapsack version; for completeness, we include it here.

Theorem 3.5.3. *Maximum Coverage subject to independent Probabilistic Set Failure is a submodular function.*

Proof. Each S_i is an element our solution can purchase. Let $E(\cdot)$ denote the objective. To prove submodularity we will establish that the law of diminishing returns holds: for an arbitrary S_i, if $A \subseteq B \subseteq S$, then

$$E(A \cup S_i) - E(A) \geq E(B \cup S_i) - E(B).$$

For a set $R \subseteq S$ which does not contain S_i we can say that the probability that S_i fails is 1. After S_i is added to either A or B the probability that S_i fails is p_i. Before S_i is "added" either: neither A or B contain S_i, or $S_i \in A$ (and $S_i \in B$), or $S_i \notin A$ and $S_i \in B$, or $S_i \notin A$ and $S_i \notin B$.

In every case, the probability that S_i fails when A is the solution is at least the probability that S_i fails when B is the solution. Letting $\wp_{S_i}(R)$ denote the probability of failure of S_i as a function of the set R:

$$\begin{aligned}&\wp_{S_i}(A) \geq \wp_{S_i}(B) \Rightarrow \\ &\wp_{S_i}(A) - \wp_{S_i}(A \cup S_i) \geq \wp_{S_i}(B) - \wp_{S_i}(B \cup S_i)\end{aligned}$$

Next, focus on a particular element y from the ground set. Either S_i does not contain y (in which case the addition of S_i leaves the expected coverage of y unchanged) or it does. For the later case, the probability of every one of the other sets containing y failing (excepting S_i) is at least for A what it is for B. Let $P(R)$ denote the probability that every one of the sets containing y fails when the solution is R. Then $P(A) \geq P(B)$ gives that

$$\begin{aligned}&P(A)(\wp_{S_i}(A) - \wp_{S_i}(A \cup S_i)) \geq \\ &\quad P(B)(\wp_{S_i}(B) - \wp_{S_i}(B \cup S_i)) \\ &\Rightarrow P(A)\wp_{S_i}(A) - P(A)\wp_{S_i}(A \cup S_i) \geq \\ &\quad P(B)\wp_{S_i}(B) - P(B)\wp_{S_i}(B \cup S_i). \\ &\Rightarrow (1 - P(A)\wp_{S_i}(A \cup S_i)) - (1 - P(A)\wp_{S_i}(A)) \geq \\ &\quad (1 - P(B)\wp_{S_i}(B \cup S_i)) - (1 - P(B)\wp_{S_i}(B)). \\ &\Rightarrow E(A \cup S_i) - E(A) \geq E(B \cup S_i) - E(B).\end{aligned}$$

The second to last line compares the changes in the probability that y is covered which results when S_i is added to A and when S_i is added to B. The final inequality follows from simply summing the change in expected coverage over all y in the ground set (including those for which the addition of S_i caused no changes in the probability of coverage). This establishes submodularity. □

3.6 Running CCPV

The method described in Calinescu, Chekuri, Pal and Vondrák [11] requires the ability to sample the value of the submodular function being maximized in polynomial time. Though for a given solution $A \subseteq S$ there are exponentially-many possible realizations of set failures, the objective can be computed in polynomial time by taking advantage of the independence of the set failures and the form of the coverage function. In particular, the expected size of the union of the sets realized as effective is just the sum of the expectations of indicators over all items of the ground set. For each such item, y, the probability of coverage is simply 1 minus the probability of non-coverage (the product of the failure probabilities of all sets in A containing y).

3.7 Easy extensions

- All arguments work for node-weighted versions of the problems (where the value associated with protecting nodes in the graph, or covering the elements in the ground set, can vary and is specified in the input).
- The deterministic $(1-\frac{1}{e})$-approximation that we observe for the spreading-vaccination Firefighter Problem also holds for our Stochastic Graph Vaccination model provided that the $f_j(x,t)$ are $\{0,1\}$- valued. That is: for the spreading-vaccination Firefighter Problem, allowing the number of vaccines to be nonuniform in time and allowing the possibility that each vaccine has a specified set of nodes where it does not work does not affect how

closely the objective can be approximated by a deterministic algorithm. This follows because the reduction to maximizing a maximum coverage objective subject to group-constraints still holds, allowing application of [1].

- The multi-infection-source case of spreading-vaccination Stochastic Graph Vaccination is also maximum coverage subject to probabilistic set failure (however, the multi-infection-source case of non-spreading Stochastic Graph Vaccination in trees is not submodular).

  This can be obtained by redefining S_{jxt} in the reduction for the single source case. Now, if R is the set of sources, let S_{jxt} contain all nodes y for which $d(x, y) \leq min_{r \in R}\, d(y, r) - t$. Pushing slightly further, *the sources may even arise at different times*: if, for each $r \in R$, r first becomes infectious at time t_r, then define S_{jxt} to be the set of all nodes y for which $d(x, y) \leq \min_{r \in R} \{d(y, r) - (t - t_r)\}$. Either described extension can also be quickly obtained by reduction to the single-source case of spreading-vaccination Stochastic Graph Vaccination: add a super source that acts as a single infection point which is connected by paths (of new degree-two nodes) of $t_r + 1$ edges to source r (for each $r \in R$). Let the efficacy of every vaccine at the all new nodes be 0 (and the value of all new nodes is 0) and shift all $f_j(x, t)$ to the right one time step (and let $f_j(x, 0) = 0$ for all (x, j)).

- For the case of infection from a single source: results for spreading-vaccination Stochastic Graph Vaccination also hold if the rate of vaccination spread is greater than the rate of infection spread (this is true even if the spread rates of vaccines vary, provided that each vaccine spreads faster than the infection does). This follows because the reduction to maximum coverage with probabilistic failure subject to a partition-matroid constraint still holds: the set of nodes in S_{jxt} is simply all nodes which will be reached by vaccine j starting from x at time t before they will be reached by the infection (though it is less easy to write down an analytic expression for this set, it can be computed explicitly in polynomial time). Crucially, it is still the case that a node is protected exactly when one of these sets is realized as effective (this is not true in a multi-infection-source case where vaccination is faster-spreading, or in any interesting case where infection is faster spreading than vaccination).

CHAPTER 4

IMPROVED LOWER BOUNDS FOR THE UNIVERSAL AND *A PRIORI* TSP

The work in this chapter is joint with Igor Gorodezky, Robert Kleinberg, and David Shmoys and appeared in the Proceedings of APPROX-RANDOM 2010 [22].

4.1 Summary

We consider two partial-information generalizations of the metric traveling salesman problem (TSP) in which the task is to produce a total ordering of a given metric space that performs well for a subset of the space that is not known in advance. In the universal TSP the subset is chosen adversarially, and in the *a priori* TSP it is chosen probabilistically. The universal TSP has been studied extensively starting in the mid-80's with the work of Bartholdi & Platzman and Bertsimas & Grigni, and more recently by Jia et al., Hajiaghayi et al., and Gupta et al. Though the *a priori* TSP was introduced in a 1985 PhD thesis of Jaillet, there have been few results until recent work by Schalekamp & Shmoys and Shmoys & Talwar.

We prove a lower bound of $\Omega(\log n)$ for the universal TSP by bounding the competitive ratio of shortest-path metrics on Ramanujan graphs. Our probabilistic proof is based on an analysis of random walks on these graphs and makes essential use of their spectral and combinatorial properties. Our lower bound for the universal TSP improves on the previous best bound of Hajiaghayi, Kleinberg & Leighton, who showed that the competitive ratio of the $n \times n$ grid is $\Omega(\sqrt[6]{\log n/\log\log n})$.

Next we show that for a large class of combinatorial optimization problems that includes TSP, a bound for the universal problem implies a matching bound on the approximation ratio achievable by deterministic algorithms for the corresponding black-box *a priori* problem. This theorem follows from an information-theoretic argument and provides an explicit testimonial to the power of randomization: it implies that the approximation ratios guaranteed by several existing randomized algorithms for *a priori* problems cannot be matched by deterministic algorithms. As a consequence, our lower bound of $\Omega(\log n)$ for the universal TSP implies a matching lower bound for the black-box *a priori* TSP. This shows that independence is critical to Shmoys & Talwar's $O(1)$-approximation in the case of independent activation probabilities: no deterministic algorithm for the black-box case of the general problem can achieve $o(\log n)$-approximation.

4.2 Introduction

A delivery person has a fixed list of potential clients that is known in advance, but on each day only a subset of the clients must be served. Rather than reoptimizing the delivery route every day, the delivery person might seek to minimize distance traveled using the following heuristic: decide in advance on a master tour of the entire client list, and on each particular day visit the set of active clients in the order that they would be served by the master tour. How well does this heuristic perform? That is, given a list of clients residing in an arbitrary finite metric space, is there a master tour of the list such that the tour induced on an active subset of clients by shortcutting the master tour tends to be close to optimal? Naturally, the answer depends on how active subsets are generated; the two canonical models are the *adversarial* and the *probabilistic*.

In the adversarial model, the master tour is announced and an adversary chooses an active subset S that maximizes the ratio of the length of the induced tour to the length of the optimal tour of S. In this case the delivery person faces the *universal traveling salesman problem* (TSP). If for a given master tour there is a ρ such that for every active subset, the length of the induced tour is guaranteed to be no more than ρ times the length of the optimal tour, we say that this master tour achieves *competitive ratio* ρ.

Alternately, in the probabilistic model there is a fixed probability distribution over subsets of clients, and the delivery person seeks a master tour that minimizes the expected cost of the induced tour. This is known as the *a priori* TSP. In this Chapter we will focus on the *black-box a priori* model in which information about the distribution on active client sets is available only through calls to a black-box oracle. Note that both the universal and *a priori* variants of TSP are NP-hard since they are both generalizations of metric TSP.

In this Chapter we give improved lower bounds for the universal and *a priori* TSP. In particular, we exhibit a family of metric spaces on which the competitive ratio of any solution to the universal TSP is $\Omega(\log n)$, improving on the previous bound of $\Omega(\sqrt[6]{\log n/\log\log n})$ (proved for a different family of metric spaces; see details below). We then extend this lower bound to the *a priori* TSP by showing how to translate a universal lower bound into a matching lower bound on approximation for a large class of *a priori* optimization problems (that includes TSP).

Prior work. The universal TSP was originally motivated by the need for an easy and efficient vehicle routing system for the "Meals on Wheels" program in Atlanta, GA. This problem was considered in Bartholdi et al. [6], who phrased it as the universal TSP on a finite subset of the plane and suggested a heuristic based on space-filling curves. Bartholdi & Platzman proved in [5] that this heuristic yields master tours with a competitive ratio of $O(\log n)$ (not only on the plane but in Euclidean space in general; see also [39]).

Bertsimas & Grigni subsequently showed in [7] that $O(\log n)$ is asymptotically tight for this particular method, and made the general conjecture that $\Omega(\log n)$ is a lower bound for the universal TSP on the plane. Working towards this conjecture, Hajiaghayi, Kleinberg & Leighton showed in [25] that any master tour of the vertices of the $n \times n$ grid has its competitive ratio bounded by $\Omega(\sqrt[6]{\log n / \log \log n})$. The main result of this Chapter (Theorem 4.4.1) is an improved lower bound of $\Omega(\log n)$ on the competitive ratio for the universal TSP, though for a different, non-planar family of metrics (see the discussion below).

Recently, there has been significant progress on algorithms for the universal TSP; inspired by [5], Jia et al. formulate in [30] the universal TSP for general finite metric spaces and give an algorithm that produces, given a metric space on n points, a master tour with competitive ratio $O(\log^4 n / \log \log n)$. Their algorithm achieves a competitive ratio of $O(\log n)$ on doubling metrics (which include constant-dimensional Euclidean metrics). At present, the best universal

TSP algorithm is due to Gupta et al. [23] (using ideas of [16]) and produces a master tour with competitive ratio $O(\log^2 n)$ on n-point metric spaces.

There has been considerably less work on the *a priori* TSP. The general notion of *a priori* optimization was introduced in Jaillet's PhD thesis [29] (see also [8]). The current best algorithmic results for the black-box *a priori* TSP are a 4-approximation due to Shmoys & Talwar [45] for the case of *independent activation* (in which points appear in the active set independently), and a randomized $O(\log n)$-approximation algorithm due to Shmoys & Schalekamp [42] for the general problem. The former algorithm depends deterministically on a single black-box sample while the latter is entirely distribution-free (that is, it requires no samples). We observe that any algorithm for the universal TSP with competitive ratio ρ can be made into a distribution-free algorithm for the *a priori* TSP that achieves a ρ-approximation. Save for the trivial observation that an inapproximability result for the classical metric TSP translates to a matching lower bound for the *a priori* TSP, no lower bounds for the *a priori* TSP were known prior to this work: a corollary to our second main result (Corollary 4.5.3) is a lower bound of $\Omega(\log n)$ on the approximation ratio of any deterministic algorithm for the black-box *a priori* TSP.

A great deal of recent work in approximation algorithms has been devoted to the design and analysis of optimization algorithms that must produce solutions when the input is only partially known, or is determined stochastically. Examples include online algorithms, stochastic optimization (see [24]), the work of Jia et al. [30] on universal optimization, as well as recent results on oblivious routing (see [23] and the references within). Our work on universal and *a priori* variants of the classical metric TSP fits squarely within this conceptual framework. Moreover, though we focus on the TSP, there are many other classical optimization problems which yield natural and interesting universal and *a priori* variants. The second main result provides a direct connection between worst-case, universal lower bounds and average-case, *a priori* lower bounds, elucidating the connection between these two problem variants not only for the TSP but for a large class of natural combinatorial optimization problems.

Main results and techniques. Recall that given a master tour of a metric space X, its competitive ratio is the maximum ratio, over all $S \subseteq X$, of the length of the induced tour of S to the length of the optimum tour of S. The competitive ratio of the universal TSP on X is the minimum of the competitive ratio over all master tours of X. In our first main result (Theorem 4.4.1) we exhibit a family of metric spaces (in particular, shortest-path metrics on Ramanujan graphs) for which the competitive ratio of the universal TSP is $\Omega(\log n)$. This proves a conjecture of Hajiaghayi, Kleinberg & Leighton from [25].

Our proof of this result is probabilistic and proceeds as follows. Let X be the shortest-path metric on a Ramanujan graph, and fix some master tour of X. We give a simple procedure for producing $S \subseteq X$ such that the length of the induced tour of S is at least $\Omega(\log n)$ times the length of the optimal tour: take a random walk on the graph and let S be the set of distinct vertices encountered by the walk. We show that with positive probability a constant fraction of the vertices in S are $\Omega(\log n)$-far from their successors with respect to the tour of S induced by the master tour. Since the length of an optimal tour of S is $O(|S|)$, this proves the desired property. Our proof makes essential use of the spectral properties of Ramanujan graphs.

Our second main result (Theorem 4.5.2) shows that for a large class of natural optimization problems that includes TSP, a lower bound on the competitive ratio of the universal version of the problem implies a matching bound on the approximation ratio of a deterministic algorithm for the black-box *a priori* version. Thus our lower bound implies that no deterministic approximation algorithm for the black-box *a priori* TSP has approximation ratio $o(\log n)$. This limit on the power of deterministic algorithms provides an interesting complement to positive results in the existing literature ([42],[45],[4]). For example, if randomization is allowed then the distribution-free algorithm of Shmoys & Schalekamp [42] (using ideas from [16]) achieves an approximation of $O(\log n)$ for the *a priori* TSP, but no deterministic algorithm that is allowed even a polynomial number of samples from the distribution can achieve a better asymptotic approximation.

4.3 Universal and *a priori* TSP

Given a metric space (X, δ) with $|X| = n$, a *master tour* is a total ordering of the points of X, which we write as a bijection $\tau : [n] \to X$. Given $x \in X$ we will refer to $\tau^{-1}(x)$, the preimage of x under τ, as its *index*. The total length of a master tour τ will be written $\delta(\tau)$:

$$\delta(\tau) = \sum_{i=1}^{n} \delta(\tau(i), \tau(i+1))$$

where we identify element $\tau(n+1)$ with element $\tau(1)$. In words, $\delta(\tau)$ is the total distance traveled when traversing X in the order given by τ (including returning, at the end of the tour, from $\tau(n)$ back to $\tau(1)$).

If $S \subseteq X$ then we write $\tau|_S$ for the tour of S obtained by *shortcutting* τ; this tour starts at the element of S of smallest index, travels to the element of S of second smallest index, etc., and at the end returns to the element of smallest index. We will write $\delta(\tau|_S)$ for the total length of this tour. Given $S \subseteq X$ let $\mathrm{opt}(S)$ denote some optimal tour of (S, δ), i.e. a tour of S that minimizes total distance traveled. The length of an optimal tour of S will be written $\delta(\mathrm{opt}(S))$. The object $\mathrm{opt}(S)$ is clearly not well-defined as there could exist multiple optimal tours, but we will ignore this technicality since our analysis is only concerned with the quantity $\delta(\mathrm{opt}(S))$.

We define $\rho(X, \delta, \tau)$ to be the competitive ratio of a master tour τ of (X, δ),

$$\rho(X, \delta, \tau) = \max_{S \subseteq X} \frac{\delta(\tau|_S)}{\delta(\mathrm{opt}(S))}$$

and define $\rho(X)$ to be the competitive ratio of the universal TSP on (X, δ),

$$\rho(X) = \min_{\tau} \rho(X, \delta, \tau) = \min_{\tau} \max_{S \subseteq X} \frac{\delta(\tau|_S)}{\delta(\mathrm{opt}(S))}.$$

We suppress the metric δ in the notation $\rho(X)$ because it will always be clear from context.

Notation: In this Chapter, (X, δ) will usually be the metric space associated with an undirected graph G; that is, X is the vertex set of G and δ is the shortest-path metric on G (the edges of G will always be unit-weighted). In this case, we write $\rho(G)$ instead of $\rho(X)$.

The universal TSP has as its input a finite metric space (X, δ) and asks to produce a master tour τ that minimizes $\rho(X, \delta, \tau)$. In the *a priori* TSP, we are given (X, δ) with a probability distribution D over 2^X. The problem asks to produce a master tour τ that minimizes $\mathbb{E}_D[\delta(\tau|_S)]$, the expected length of the induced tour of S with respect to D. In this work we consider the black-box model of the *a priori* TSP in which no information about D is part of the input, but an algorithm may sample subsets of X according to D from a black box in polynomial time.

4.4 A lower bound for the universal TSP

In this section we prove a lower bound for the universal TSP on Ramanujan graphs. In order to define these precisely and formally state our result, we must make a few additional definitions. Given an $n \times n$ symmetric matrix M, we denote its (real) eigenvalues, in decreasing order, by $\lambda_i(M)$ for $i = 1, \ldots, n$. Recall that if G is an undirected d-regular graph, A is its adjacency matrix, and $M = \frac{1}{d}A$ is the transition matrix for a random walk on G, then $\lambda_1(M) = 1$. Let us refer to the eigenvalues of the transition matrix M as the eigenvalues of the graph G, written $1 = \lambda_1(G) \geq \lambda_2(G) \geq \cdots \geq \lambda_n(G)$. We say that a d-regular graph G is a *Ramanujan graph* if

$$|\lambda_i(G)| \leq 2\frac{\sqrt{d-1}}{d} \tag{4.1}$$

for $i \geq 2$ (i.e. if the Alon-Boppana bound [37] is asymptotically tight for G).

Now we may state our main result. Recall that the *girth* of G is the length of its shortest cycle.

Theorem 4.4.1. *For all sufficiently large d there exists n_0 such that if G is a d-regular*

Ramanujan graph on $n \geq n_0$ vertices whose girth is at least $\frac{2}{3}\log_{d-1} n$, then $\rho(G) = \Omega(\log n)$.

Before moving on to the proof of this theorem, we note that its hypotheses are an explicit reference to the Ramanujan graphs constructed by Lubotzky, Phillips & Sarnak in [34]. In that paper, the authors prove that for every odd prime p there exists an infinite family of $(p+1)$-regular Ramanujan graphs whose girth is at least $\frac{2}{3}\log_p n$ (this construction was later extended by Morgenstern in [36] to allow prime powers). It is well-known that if G is a d-regular graph on n vertices then its girth can be at most $2\log_{d-1} n$ (see, for instance, Section IV.1 of [9]), so that the girth of the Lubotzky-Phillips-Sarnak graphs is very nearly as large as possible. As we will see, this large-girth property will be central to the proof of Theorem 4.4.1.

The strategy: a probabilistic proof.

We will prove Theorem 4.4.1 by using the probabilistic method. Namely, in this section we will reduce the theorem to a claim about random walks on Ramanujan graphs (then prove this claim in subsequent sections). We shall require several lemmas analyzing random walks on graphs of large girth; these lemmas and their proofs are relegated to Section 4.7.1.

Let G be a d-regular Ramanujan graph on the vertex set V, with $|V| = n$, and let δ be the shortest-path metric on G. Given a master tour τ of the metric space (V, δ), let us use the term *bad set* for a set $S \subset V$ such that $\delta(\tau|_S)/\operatorname{opt}(S) = \Omega(\log n)$. We would like to prove that a bad set exists for every master tour of G, which would imply $\rho(G) = \Omega(\log n)$. Our strategy is as follows.

Fix a master tour τ of G, let g be the girth of G (recall that one of the assumptions in Theorem 4.4.1 is that g is large), and consider uniformly sampling a random walk of $L = 70g$ steps; that is, choose a starting vertex uniformly at random and perform a random walk for L steps. We regard an L-step random walk as a map $\mathsf{W} : [L] \to [n]$ where $\mathsf{W}(i)$ is the index of the ith vertex in the walk. Note that $\mathsf{W}(i)$ refers to an index rather than a vertex; the ith vertex of W would be written $\tau(\mathsf{W}(i))$.

Define V_{W} to be the set of distinct vertices encountered by W:

$$V_{\mathsf{W}} = \{\tau(\mathsf{W}(i)) \mid i = 1, \ldots, L\}.$$

It is easy to see that $\delta(\mathrm{opt}(V_{\mathsf{W}})) = O(L)$, since walking along W and then back again produces a tour of V_{W}. If we can show that there exists a V_{W} for which $\delta(\tau|_{V_{\mathsf{W}}}) = \Omega(L \log n)$ then we are done, for such a V_{W} is a bad set. We will prove the existence of such a V_{W} using probabilistic arguments.

To this end, given a walk W and an index $\mathsf{W}(i)$ define the *successor* of $\mathsf{W}(i)$ (with respect to τ) to be

$$\mathsf{W}_{succ}(i) = \min\{\mathsf{W}(j) \mid \mathsf{W}(i) < \mathsf{W}(j)\}.$$

That is, $\mathsf{W}_{succ}(i)$ is the index of the vertex in V_{W} that follows $\mathsf{W}(i)$ when V_{W} is traversed by shortcutting τ (as usual, the successor of the vertex in V_{W} with largest index is the vertex with the smallest index).

Theorem 4.4.2. *Let G be a d-regular Ramanujan graph on n vertices with girth $g \geq \frac{2}{3}\log_{d-1} n$, fix a master tour of G, and set $L = 70g$. For all sufficiently large d there exists n_0 such that if $n \geq n_0$ and* W *is a uniformly sampled L-step random walk on G,* *then*

$$\Pr\left[\delta\big(\tau(\mathsf{W}(i)),\tau(\mathsf{W}_{succ}(i))\big) > \frac{2}{3d}\log_{d-1} n\right] = \Omega(1)$$

for every $i = 1, 2, \ldots, g/2$, where δ is the shortest-path metric on G.

Before proving this theorem, let us see how it implies the existence of a V_{W} that is a bad set. Given an L-step walk W, let $X(\mathsf{W})$ be the number of events of the form

$$\delta\big(\tau(\mathsf{W}(i)),\tau(\mathsf{W}_{succ}(i))\big) > \frac{2}{3d}\log_{d-1} n, \text{ and } \mathsf{W} \text{ visits vertex } \tau(\mathsf{W}(i)) \text{ at most } 140 \text{ times} \tag{4.2}$$

The key observation is that we can bound $\delta(\tau|_{V_{\mathsf{W}}})$ from below by

$$\delta(\tau|_{V_{\mathsf{W}}}) \geq \left(\frac{X(\mathsf{W})}{140}\right)\left(\frac{2}{3d}\log_{d-1} n\right) = \left(\frac{1}{210d\log(d-1)}\right)X(\mathsf{W})\log n$$

because traversing V_{W} according to τ means making at least $X(\mathsf{W})/140$ trips of length at least $\frac{2}{3d}\log_{d-1} n$. We will use Theorem 4.4.2 to prove the existence of a W such that $X(\mathsf{W}) = \Omega(L)$, which by the above would prove the existence of a bad V_{W}.

For brevity, let us write the events "$\delta\big(\tau(\mathsf{W}(i)),\tau(\mathsf{W}_{succ}(i))\big) > \frac{2}{3d}\log_{d-1} n$" and "$\tau(\mathsf{W}(i))$ is visited at most 140 times by W" as E_i and E'_i, respectively. Now, $\mathbb{E}[X(\mathsf{W})]$ is the sum of the probabilities of $E_i \cap E'_i$:

$$\mathbb{E}[X(\mathsf{W})] = \sum_{i=1}^{L}\Pr[E_i \cap E'_i] \geq \sum_{i=1}^{g/2}\Pr[E_i \cap E'_i]. \tag{4.3}$$

First note that $\Pr[E_i \cap E'_i] \geq \Pr[E_i] - \Pr[\overline{E'_i}]$. Theorem 4.4.2 says that there exists a constant A such that $\Pr[E_i] \geq A$ when $i \leq g/2$. Now, by Lemma 4.7.5 from the technical addendum, $\Pr[\overline{E'_i}]$ can be made arbitrarily small by taking d and then n large enough (hence this requirement in the statement of Theorem 4.4.1).

Let us, then, take d and n large enough so that $\Pr[\overline{E_i'}] \leq A/2$. In light of all this, equation (4.3) now gives

$$\mathbb{E}[X(\mathsf{W})] \geq \sum_{i=1}^{g/2} \left(\Pr[E_i] - \Pr[\overline{E_i'}]\right) \geq \frac{A}{4} g = \frac{A}{280} L.$$

Since $\mathbb{E}[X(\mathsf{W})] = \Omega(L)$, there must exist a walk $\overline{\mathsf{W}}$ for which $X(\overline{\mathsf{W}}) = \Omega(L)$, meaning the event (4.2) holds for a constant fraction of the L steps of the walk. As noted in the discussion above, this implies $\delta(\tau|_{V_{\overline{\mathsf{W}}}}) = \Omega(L \log n)$, which shows that $V_{\overline{\mathsf{W}}}$ is a bad set. Thus we find that Theorem 4.4.2 implies Theorem 4.4.1, so it remains to prove the former.

Proving Theorem 4.4.2.

This section is dominated by technical arguments concerning random walks on graphs. In order to present these as clearly as possible, we introduce some notation. Recall that Theorem 4.4.2 is a statement about a random walk W on a Ramanujan graph; we shall reserve this notation for this particular random walk, owing to its central role in the proof of Theorem 4.4.1 (as discussed in the section on probabilistic proof strategy). Our proof of Theorem 4.4.2, however, will require general lemmas about random walks on regular graphs, which we will later apply to W, or more often sub-walks of W. In these lemmas we will use the notation w for an arbitrary random walk. As before, $w(i)$ is the index of the vertex at step i of w, and $\tau(w(i))$, where τ will always be some fixed master tour, is the vertex at step i.

One last crucial piece of notation: given $j \in [n]$ and a positive integer $\ell \leq n$, define

$$[j : \ell]_n = \{(j+1) \bmod n, (j+2) \bmod n, \ldots, (j+\ell) \bmod n\}.$$

This notation will appear in the following context: fix a master tour τ of a graph G and consider a random walk w on G. Then $w(i)$ is the index of the vertex at step i of w, and $[w(i) : \ell]_n$ consists of the ℓ indices that follow $w(i)$ in τ, with wraparound modulo n.

The proof of Theorem 4.4.2 requires the following technical lemma.

Lemma 4.4.3. *Let G be a d-regular graph on n vertices such that $|\lambda_i(G)| \leq \eta$ for $i \geq 2$, say $\eta = .05$. Fix a master tour of $V(G)$, some integer k, and set $\ell = \lceil n/(32k) \rceil$. If w is a uniformly sampled k-step random walk on G, then the probability that w encounters a vertex whose index is in $[w(1) : \ell]_n$ is less than $1/4$.*

We relegate the proof of Lemma 4.4.3 to Section 4.7.2 of the technical addendum, and proceed to the proof of Theorem 4.4.2.

Proof of Theorem 4.4.2. Recall the setting of Theorem 4.4.2: we have a d-regular Ramanujan graph G on n vertices whose girth is $g \geq \frac{2}{3} \log_{d-1} n$. We fix a master tour of G and uniformly sample an L-step random walk W, where $L = 70g$. We want to show that for every $i = 1, 2, \ldots, g/2$ we have

$$\Pr\left[\delta\big(\tau(\mathsf{W}(i)), \tau(\mathsf{W}_{succ}(i))\big) > \frac{2}{3d} \log_{d-1} n\right] = \Omega(1).$$

Fix, then, some $i \in \{1, \ldots, g/2\}$. We can regard W as two independent random walks w_1, w_2 that both begin at $\tau(\mathsf{W}(i))$, i.e. $w_1(1) = w_2(1) = \mathsf{W}(i)$. See Figure 4.1. Note that $\tau(\mathsf{W}(i))$ is uniformly sampled from V since W itself is a uniformly sampled walk. Let L_1, L_2 be the number of steps of w_1 and w_2, respectively. Since $i \leq g/2$ we know that $L_1 \leq g/2$ and $\frac{139}{2} g \leq L_2 \leq 70g$.

First we apply Lemma 4.4.3 to w_1 (with $k = g/2$) and then once again to the first $g/2$ steps of w_2. Observe that this requires choosing d large enough so that $|\lambda_i(G)| \leq .05$ for $i \geq 2$, which is possible since G is a Ramanujan graph (recall inequality (4.1)). Now, combining these two applications of Lemma 4.4.3 with the union bound, we find that with probability at least $1/2$ both of the following events hold :

(i) w_1 doesn't encounter any of the $\ell = \lceil n/(16g) \rceil$ indices that follow $w_1(1) = \mathsf{W}(i)$;

(ii) in its first $g/2$ steps, w_2 doesn't encounter these indices either.

Now, we apply Lemma 4.7.6 to the walk w_2 (use $k = 140$), and choose n large enough so that the probability of the two events

(iii) in its first $g/2$ steps, w_2 leaves the g/d-ball around $\tau(\mathsf{W}(i))$, and

(iv) w_2 never returns to this g/d-ball in subsequent steps

is at least $9/10$. Putting it all together via the union bound, we find that all four of the events $(i) - (iv)$ hold with probability at least $1 - 1/2 - 1/10 = 2/5$. Next, note that there are at least $69g$ steps left in w_2 after the first $g/2$. We claim that with probability $4/5$,

(v) one of the last $64g$ steps of w_2 contains one of the $\lceil n/(16g) \rceil$ indices that follow $w_2(1) = \mathsf{W}(i)$.

This claim will complete the proof: if event (v) holds along with events $(i) - (iv)$ (which happens with probability at least $1 - 3/5 - 1/5 = 1/5$ by the union bound) then $\mathsf{W}_{succ}(i)$, which is the next-largest (modulo n) index in W after $\mathsf{W}(i)$,

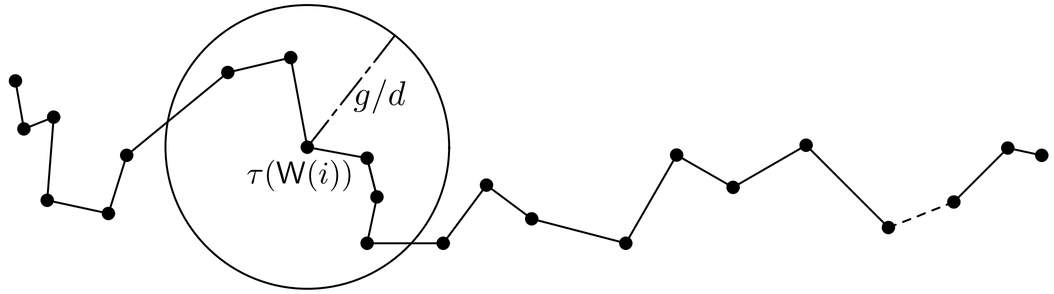

Figure 4.1: The g/d-ball centered at $\tau(\mathsf{W}(i))$. The walk w_1 is drawn to the left of $\tau(\mathsf{W}(i))$ and the much longer walk w_2 is drawn to the right.

must be one of the $\lceil n/(16g) \rceil$ indices that follow $\mathsf{W}(i)$. But none of these indices are encountered by W inside the g/d-ball around $\tau(\mathsf{W}(i))$, which implies $\delta\big(\tau(\mathsf{W}(i)), \tau(\mathsf{W}_{succ}(i))\big) > \frac{2}{3d} \log_{d-1} n$.

The claim about (v) is motivated by the following intuition. Because our graph is a strong expander, w_2 rapidly mixes in its first $\frac{11}{2} g$ steps. Therefore the last $64g$ steps approximate a stationary random walk on G. Since the indices that follow $w_2(1)$ are a $(16g)^{-1}$-fraction of the vertices and stationary random walks resemble independent sampling, we would expect a $64g$-step, almost stationary random walk to hit this index set with large probability. This argument is formalized in the technical addendum as Lemma 4.7.7. □

4.5 Universal TSP bound implies *a priori* TSP bound

Using a framework for abstract optimization (similar to [30], [24] for, respectively, universal and stochastic optimization) we show how to convert universal lower bounds into matching *a priori* lower bounds on deterministic algorithms for a large and natural class of optimization problems that contains TSP. In order to be applicable to the TSP, our definitions describe minimization problems, but analogous definitions and results hold for maximization problems.

Abstract optimization problems.

An abstract combinatorial optimization problem is a triple $\Pi = (X, c, M)$ where X is a universe of *clients* and c, M are functions defined as follows.[1] With each $S \subseteq X$ we associate a collection of *feasible solutions* that we denote $\mathrm{sol}(S)$. The function c is a *cost function* that maps, for every $S \subseteq X$, solutions in $\mathrm{sol}(S)$ to $\mathbb{R}$. The function M is a *restriction map* that given $S' \subseteq S$ and $F \in \mathrm{sol}(S)$ outputs a feasible solution for S': $M(F, S') \in \mathrm{sol}(S')$. When $S = X$ with $F \in \mathrm{sol}(X)$, we will use the notation $F|_{S'}$ for $M(F, S')$, and say that $F|_{S'}$ is the master solution F *restricted* to S', or that $F|_{S'}$ is the solution for S' *induced* by F.

Given $S \subseteq X$ we use $\mathrm{opt}(S)$ to denote a feasible solution for S of minimum cost. The *classical* version of Π specifies $S \subseteq X$ and asks to produce $F \in \mathrm{sol}(S)$ of minimum cost. The *universal* version of Π asks for $F \in \mathrm{sol}(X)$ that minimizes the *competitive ratio*

$$\rho(X, F) = \max_{S \subseteq X} \frac{c(F|_S)}{c(\mathrm{opt}(S))}.$$

We write $\rho(\Pi)$ for the competitive ratio of Π

$$\rho(\Pi) = \min_{F \in \mathrm{sol}(X)} \left(\max_{S \subseteq X} \frac{c(F|_S)}{c(\mathrm{opt}(S))} \right)$$

and use the notation $\mathrm{opt}_u(X)$ for a solution that is optimal with respect to this universal objective, i.e. for which $\rho(X, \mathrm{opt}_u(X)) = \rho(\Pi)$.

[1] Regard c and M as polynomial-time algorithms that are part of the description of Π in the form of a Turing machine encoding.

In the *a priori* version of Π, there is a distribution D on subsets of X. The *a priori* objective is to return $F \in \mathrm{sol}(X)$ that minimizes $\mathbb{E}_D[F|_S]$. We denote a solution optimal with respect to this objective by $\mathrm{opt}_{ap}(X, D)$, so that if $p(S)$ is the probability given to $S \subseteq X$ by D, then

$$c(\mathrm{opt}_{ap}(X, D)) = \min_{F \in \mathrm{sol}(X)} \sum_{S \subseteq X} c(F|_S)p(S).$$

We will focus on the *black-box* model of *a priori* optimization in which no information about D is part of the input describing the *a priori* version of Π, but an algorithm may sample subsets of X according to D from a black box in polynomial time.

Notice that the definitions in this section generalize the content of Section 4.3: in the case of TSP, X is a set of points, feasible solutions to $S \subseteq X$ are total orderings of S, the cost function c encodes the metric on X, and M restricts tours by shortcutting.

Approximate solutions to *a priori* problems.

There are two key definitions of approximate *a priori* solutions that feature in the literature. Given the *a priori* optimization problem (Π, D), an algorithm A is a *hard β-approximation* if A runs in polynomial time (in the description of Π), makes polynomially-many black-box queries, and produces a solution $F \in \mathrm{sol}(X)$ such that

$$\Pr\left[\frac{\mathbb{E}_D[F|_S]}{\mathbb{E}_D[\mathrm{opt}_{ap}(X, D)|_S]} \leq \beta\right] > \frac{1}{2} \tag{4.4}$$

where the probability is over the black-box output as well random choices made by A.

We say that an algorithm A is a *soft β-approximation* if A runs in polynomial time, makes polynomially-many black-box queries, and produces a solution $F \in \mathrm{sol}(X)$ such that

$$\mathbb{E}\left[\frac{\mathbb{E}_D[F|_S]}{\mathbb{E}_D[\mathrm{opt}_{ap}(X, D)|_S]}\right] \leq \beta \tag{4.5}$$

where the expectation is over the black-box output as well random choices made by A.

The notion of hard vs. soft approximation is analogous to the distinction between Monte Carlo and Las Vegas algorithms. The analogy extends, via Markov's inequality, to the following lemma.

Lemma 4.5.1. *Any soft β-approximation to an a priori minimization problem is a hard 2β-approximation.*

We say that A is a *deterministic* hard or soft β-approximation if it is deterministic in the black box output. In this case the outermost probability in inequality (4.4) and the outermost expectation in inequality (4.5) depend only on the black-box output. On the other hand, we may explicitly say that A is randomized when we want to stress this fact. As an application to the *a priori* TSP, recall the deterministic soft 4-approximation in the case of independent activation given in [45]. Lemma 4.5.1 shows that this algorithm is also a deterministic hard 8-approximation in this case.

Worst-case bounds to average-case bounds.

In this section we prove our second main result, which shows how to convert universal lower bounds into *a priori* lower bounds on deterministic hard approximations for a certain class of optimization problems.

Our result applies to the class of optimization problems $\Pi = (X, c, M)$ that satisfy three properties:

• *Non-negative costs*: For all $S \subseteq X$, $c(F) \geq 0$ for every $F \in \mathrm{sol}(S)$.

• *Trivial set*: There exists a (possibly empty) *trivial* $T \subseteq X$ such that $c(F|_T) = 0$ for all $F \in \mathrm{sol}(X)$.

• *Optimal sub-solutions are recoverable*: For any $S \subseteq X$ there exists some $F \in \mathrm{sol}(X)$ such that M maps F to $F' \in \mathrm{sol}(S)$ with $c(F') = c(\mathrm{opt}(S))$.

Problems satisfying these axioms will be called *regular*. The reader can verify that TSP is a regular problem. It is easy to see that most of the examples given in [8] are regular, including Steiner Tree and Max Cut.

Theorem 4.5.2. *If* Π *is a regular minimization problem with competitive ratio* $\rho(\Pi) \geq \gamma$ *then no deterministic hard approximation for the a priori version of* Π *can have approximation ratio better than* γ*.*

Proof. We prove the contrapositive: if there exists a deterministic hard β-approximation algorithm A for the *a priori* version of Π then $\rho(\Pi) \leq \beta$. Since A is a hard β-approximation, there exists $\alpha < 1/2$ such that

$$\Pr\left[\frac{\mathbb{E}_D[F|_S]}{\mathbb{E}_D[\mathrm{opt}_{ap}(X, D)|_S]} \leq \beta\right] \geq 1 - \alpha. \tag{4.6}$$

Let $f(|\Pi|)$ be the number of black-box samples queried by A. Choose $\delta > 0$

sufficiently small so that $1 - \alpha - \delta > 0$ and set $k = \lceil \log\left(f(|\Pi|)/\delta\right) \rceil$ so that

$$\left(1 - \frac{1}{2^k}\right)^{f(|\Pi|)} \geq 1 - \frac{f(|\Pi|)}{2^k} \geq 1 - \delta. \tag{4.7}$$

Let T be a trivial subset (as in the definition of a regular optimization problem). Since A is deterministic, when it observes problem instance Π and $f(|\Pi|)$ samples which are each T, it must return some specific $\hat{F} \in \mathrm{sol}(X)$. We claim that with probability at least $1 - \delta - \alpha > 0$, $\hat{F}$ has competitive ratio $\rho(X, \hat{F}) \leq \beta$. Proving this claim will imply $\rho(X) \leq \beta$ and complete the proof.

To prove the claim, construct an instance of *a priori* Π by defining a distribution D as follows: choose $R \subseteq X$ that maximizes $\frac{c(\hat{F}|_R)}{c(\mathrm{opt}(R))}$, i.e. that achieves $\rho(X, \hat{F})$, and set

$$\Pr_D[S] = \begin{cases} 1 - \frac{1}{2^k} & \text{if } S = T \\ \frac{1}{2^k} & \text{if } S = R. \end{cases}$$

We have chosen k so that when we run A on (Π, D) the probability that the black-box queries made by A return a non-T subset is at most δ (see inequality (4.7)). Therefore, when we run A on (Π, D) it follows from the union bound (and the fact that A is a hard β-approximation) that the following happens with probability at least $1 - \delta - \alpha > 0$: the collection of black-box samples contains no non-T subset, and A produces a β-approximate solution.

What are the consequences of this event? Since each sample observed is T, and the problem instance is Π, A must return $\hat{F}$ (since A is deterministic). Let $\mathrm{opt}_{ap}(X, D)$ denote the optimal *a priori* solution in the instance (Π, D). Since A

gave a β-approximate solution in this case, we have (cf. inequality 4.6)

$$\mathbb{E}_D[c(\hat{F}|_S)] \leq \beta\,\mathbb{E}_D[c(\mathrm{opt}_{ap}(X,D)|_S)$$
$$\Longrightarrow \Pr(S=T)\cdot 0 + \Pr(S=R)c(\hat{F}|_R) \leq \beta\left(\Pr(S=T)\cdot 0 + \Pr(S=R)c(\mathrm{opt}_{ap}(X,D)|_R)\right)$$
$$\Longrightarrow c(\hat{F}|_R) \leq \beta c(\mathrm{opt}_{ap}(X,D)|_R).$$

Consider $\mathrm{opt}_{ap}(X,D)|_R$: all tours perform the same on the active set T (by the second regularity axiom), so $\mathrm{opt}_{ap}(X,D)$ is any solution in $\mathrm{sol}(X)$ which has minimal cost when restricted to a solution on R via M, i.e. $c(\mathrm{opt}_{ap}(X,D)|_R) = c(\mathrm{opt}(R))$. Note that such a master solution must exist since Π is regular, hence optimal sub-solutions are recoverable.

We conclude that $c(\hat{F}|_R) \leq \beta c(\mathrm{opt}(R))$, so that $c(\hat{F}|_R)/c(\mathrm{opt}(R)) \leq \beta$. Since we chose R to maximize this ratio, the bound holds for every $Q \subseteq X$:

$$\frac{c(\hat{F}|_Q)}{c(\mathrm{opt}(Q))} \leq \beta.$$

Thus, with probability at least $1-\delta-\alpha > 0$, the master solution $\hat{F}$ has competitive ratio β for the universal version of Π. In particular, there exists a master solution for universal Π with this competitive ratio, hence $\rho(\Pi) \leq \beta$. □

Combining Theorem 4.5.2 with our lower bound on the competitive ratio for universal TSP (Theorem 4.4.1), we immediately get the following corollary.

Corollary 4.5.3. *No deterministic (hard or soft) approximation algorithm for a priori TSP has approximation ratio $o(\log n)$.*

Note that though the bound in Theorem 4.5.2 applies only to hard approximation algorithms, it follows from Lemma 4.5.1 that the claim in Corollary 4.5.3

applies to soft approximations as well.

Deterministic vs. randomized approximations.

We can use Theorem 4.5.2 to translate universal lower bounds into lower bounds on deterministic *a priori* approximations for other regular optimization problems. Applying the theorem to the Steiner Tree problem illustrates that randomized *a priori* approximations can sometimes perform dramatically better than deterministic ones.

In the *a priori* Steiner Tree problem, we have a graph $G = (V, E)$ and a distribution D over 2^V. A feasible solution for $U \subseteq V$ is a subtree of G that spans U. The solution induced by $U' \subseteq U$ and some $T \in \mathrm{sol}(U)$ is the smallest connected subtree of T that spans U'. The *a priori* objective is to minimize the expected cost of the induced subtree. To bound the competitive ratio of universal Steiner tree, let G be the n-cycle with all edges of unit length. A master solution T is a spanning tree of G, hence is simply G with some edge (u, v) omitted. Setting $U = \{u, v\}$, the sub-solution on U induced by T has length $n - 1$, but the optimal solution on U has length 1. Thus the competitive ratio of any master solution is $\Omega(n)$.

A randomized algorithm can do much better: simply sample a tree-metric approximation of the shortest-path metric on G using the algorithm of Fakcharoenphol, Rao & Talwar [16]. Since the expected distortion is $O(\log n)$, this randomized (distribution-free!) algorithm achieves a soft approximation of $O(\log n)$ (the analysis is identical to the proof of Corollary 4 in [42]).

Corollary 4.5.4. *No deterministic (hard or soft) approximation algorithm for a priori Steiner Tree has approximation ratio $o(n)$, whereas there exists a randomized, distribution-free soft $O(\log n)$-approximation.*

Returning briefly to TSP, Schalekamp & Shmoys give in [42] a randomized soft $O(\log)$-approximation algorithm for *a priori* TSP that is distribution-free. Corollary 4.5.3 shows that no polynomial number of black-box samples can help a deterministic algorithm asymptotically beat this randomized approximation guarantee. In the independent activation case of *a priori* TSP, Shmoys & Talwar give in [45] a deterministic soft 4-approximation which relies on a single black-box sample. Our results show that without the assumption of independence, no polynomial number of samples is enough to help a deterministic algorithm achieve a $o(\log n)$ approximation ratio.

We end with a comment on the Euclidean case. In light of Arora's PTAS for the Euclidean TSP [4], one might hope that an $O(1)$-approximation is possible for the Euclidean *a priori* TSP in the black-box case. The universal lower bound of $\Omega(\sqrt[6]{\log n/\log\log n})$ in the case of the $n \times n$ planar grid ([25]) combined with Theorem 4.5.2 implies that for deterministic algorithms, the answer is no.

4.6 Open problems

The conjecture of Bertsimas & Grigni [7] that the lower bound of $\Omega(\log n)$ holds even for finite subsets of the plane remains open. The best known algorithm for the universal TSP (due to Gupta, Hajiaghayi & Räcke [23]) produces, for an arbitrary metric space (X, δ) on n points, a master tour τ with $\rho(X, \delta, \tau) =$

$O(\log^2 n)$. Thus the best lower and upper bounds are separated by a factor of $\log n$.

Is either bound tight? In [30], Jia et al. give an algorithm that produces a master tour τ with $\rho(X, \delta, \tau) = O(\log n)$ when (X, δ) is a constant-dimension Euclidean or bounded-growth metric. Schalekamp & Shmoys observe in [42] that a straightforward application of the tree-metric embedding scheme of Fakcharoenphol et al. [16] gives a randomized algorithm that produces a master tour whose expected competitive ratio for any fixed $S \subseteq X$ is $O(\log n)$. We conjecture that this expected guarantee for a fixed S can be matched by a deterministic guarantee for all S.

Conjecture 4.6.1. *There is a deterministic, polynomial-time algorithm that, given a metric space (X, δ) with $|X| = n$, produces a master tour τ with $\rho(X, \delta, \tau) = O(\log n)$.*

Several interesting avenues for future work are also suggested by Theorem 4.5.2. A natural question is whether any general relationship holds between universal lower bounds and *a priori* approximation ratios for randomized algorithms. If the number of samples is allowed to depend on, for example, the inverse probability of the least commonly occurring active set, is it still possible to prove some kind of universal-to-*a-priori* lower bound translation in the spirit of Theorem 4.5.2?

We note that in the case of 2-stage build-with-recourse stochastic optimization models (see [24]), most algorithms yielding a constant approximation ratio rely on either nice properties of the objective function (such as submodularity) or strong bounds on the inflation of the cost function in the recourse stage;

could Theorem 4.5.2 be extended to a larger class of algorithms by restricting to a smaller class of problems?

4.7 Technical Addendum

4.7.1 Random walks on graphs with large girth

In this section we prove several useful lemmas concerning random walks on regular graphs of large girth. Consider an L-step random walk on a d-regular graph G of girth g that begins at some fixed vertex v. Let us assume $d \geq 7$ for convenience. If we also assume $L \leq g/2$ then the L-neighborhood of v is identical to the L-neighborhood of a vertex in the infinite d-regular tree. Therefore, to analyze such a random walk on G it suffices to analyze a random walk of the same number of steps on a d-regular tree. Let us do precisely this. We shall require the following familiar version of the lower-tail Chernoff bound (see [12]).

Lemma 4.7.1. *Let $X_1, \ldots X_N$ be i.i.d. Bernoulli random variables with $\Pr[X_i = 1] = p$. If $X = \sum_i X_i$ and $0 < \epsilon < 1$ then $\Pr[X \leq \epsilon pN] \leq e^{-(1-\epsilon)^2 pN/2}$.*

We illustrate the method with an easy question: how likely is it that an L-step random walk will end reasonably far from where it began?

Lemma 4.7.2. *Consider an L-step random walk that begins at some vertex v of the infinite d-regular tree, $d \geq 7$. The probability that the walk ends at distance at most $L/3$ from v is at most $e^{-\Omega(L)}$.*

Proof. Let q be the probability that we are looking to bound. Consider a DFS representation of the L-neighborhood of v, in which v is at the top and the rest of the tree extends downward. The walk ends at distance at most $L/3$ from v if and only if the number of downward steps is at most $\frac{2}{3}L$. At any point in the walk, the probability of a downward step is $(d-1)/d$, while the probability of an upward step is $1/d$.[2] Therefore, setting $p = (d-1)/d$, q is the probability that in tossing L coins in which the probability of heads is p, at most $\frac{2}{3}L$ come up heads. The Chernoff bound (Lemma 4.7.1) combined with the observation that $d \geq 7$ implies $p \geq 6/7$ now prove the claim. □

Consider again an L-step random walk that begins at v. What is the probability that the walk never returns to v? We call this the *escape probability* $e(d, L)$ (our notation reflects this quantity's dependence on d and L). To bound $e(d, L)$, we will require the following elementary fact whose proof can be found in [41] (see Example 3.25 of that book).

Lemma 4.7.3. *Fix positive integers h and t with $h > t$, and consider the uniform distribution over the set of all sequences of $h + t$ coin tosses in which there are h heads and t tails. Then $(h - t)/(h + t)$ is the probability that for such a sequence, every initial segment contains more heads than tails.*

Lemma 4.7.4. *For an L-step random walk that begins at some vertex v of the infinite d-regular tree, $d \geq 7$, we have*

$$e(d, L) \geq 1 - 2\left(\frac{1}{d} + e^{-\Omega(L)}\right).$$

[2] This is not precisely true, since the probability that the first step is away from v is 1, but this only improves the odds of stopping far from v, though only negligibly. We will ignore this technicality.

Proof. As before, consider a DFS representation of the L-neighborhood of v. It is easy to see that the random walk never returns to v if and only if at every point in the walk, the number of downward steps away from v is strictly larger than the number of upwards steps towards v. Once again the probability of a downward step at any point in the walk is $(d-1)/d$, while the probability of an upward step is $1/d$.[3] Therefore, setting $p = (d-1)/d$, $e(d, L)$ is the probability that in tossing L coins in which the probability of heads is p, the sequence of tosses is such that every initial segment of the sequence contains more heads than tails. Let us assume that L is odd, for notational convenience. Then, using Lemma 4.7.3, we have that

$$\begin{aligned}
e(d, L) &= \sum_{k=\frac{L+1}{2}}^{L} \Pr\left[k \text{ heads, } L-k \text{ tails}\right] \cdot \left(\frac{2k-L}{L}\right) \\
&= \sum_{k=\frac{L+1}{2}}^{L} \binom{L}{k} p^k (1-p)^{L-k} \left(\frac{2k-L}{L}\right) \\
&= 2 \sum_{k=\frac{L+1}{2}}^{L} \frac{k}{L} \binom{L}{k} p^k (1-p)^{L-k} - \sum_{k=\frac{L+1}{2}}^{L} \binom{L}{k} p^k (1-p)^{L-k} \\
&= 2p \sum_{k=\frac{L+1}{2}}^{L} \binom{L-1}{k-1} p^{k-1} (1-p)^{L-k} - \sum_{k=\frac{L+1}{2}}^{L} \binom{L}{k} p^k (1-p)^{L-k} \\
&= 2p \sum_{k=\frac{L-1}{2}}^{L-1} \binom{L-1}{k} p^{k} (1-p)^{L-1-k} - \sum_{k=\frac{L+1}{2}}^{L} \binom{L}{k} p^k (1-p)^{L-k}.
\end{aligned}$$

The second term above is the probability that L tosses of a p-biased coin contain at least $(L+1)/2$ heads, hence is at most 1. The first term, in turn, is $2p$ times the probability that $L-1$ tosses of a p-biased coin contain at least $(L-1)/2$ heads.

[3] Again, this is not true for the first step, but again we ignore this technicality since it does not affect the proof.

If we call this probability q, then we have

$$e(d, L) \geq 2pq - 1.$$

But it follows immediately from Lemma 4.7.1, the Chernoff bound, that $q \geq 1 - e^{-\Omega(L)}$. Therefore

$$\begin{aligned} e(d, L) &\geq 2pq - 1 \\ &\geq 2p\left(1 - e^{-\Omega(L)}\right) - 1 \\ &= 2\left(1 - \frac{1}{d}\right)\left(1 - e^{-\Omega(L)}\right) - 1 \\ &\geq 1 - 2\left(\frac{1}{d} + e^{-\Omega(L)}\right) \end{aligned}$$

as desired. □

From regular trees to graphs with large girth

Let us recall our original motivation: L-step random walks on a d-regular graph G whose girth g is large. First consider, as before, an L-step random walk with $L \leq g/2$. For any vertex u encountered by this walk, the probability that u is encountered more than once is bounded by

$$1 - e(d, L/2) \leq 2\left(\frac{1}{d} + e^{-\Omega(L)}\right)$$

since we can condition without loss of generality on u being first encountered in the first half of the walk. We require a minor extension of this fact, specialized to the case of logarithmic girth.

Lemma 4.7.5. *Let G be a d-regular graph, $d \geq 7$, with n vertices and girth $g = \Omega(\log_{d-1} n)$. Consider an L-step random walk on G, where $L = kg/2$ for some in-*

teger k, and let u be a vertex encountered during the walk. The probability that u is encountered more than k times is at most $2k\left(\frac{1}{d}+n^{-\Omega(1)/\ln(d-1)}\right)$.

Proof. Split this walk into k consecutive walks of $g/2$ steps, call them $w_1, \ldots, w_k$. If u is encountered more than k times in w then it must have been encountered more than once in some w_i, but the probability of this event is bounded by $2\left(\frac{1}{d}+e^{-\Omega(g)}\right)$ for each i. Now the union bound gives

$$\Pr(u \text{ encountered more than } k \text{ times}) \leq 2k\left(\frac{1}{d}+e^{-\Omega(g)}\right)$$

and finally our assumption on g completes the proof. □

The lesson of the previous lemma is that because $(g/2)$-balls around vertices are tree-like, we are able to prove nontrivial claims about random walks in graphs of girth g even when those walks are longer than $g/2$ (as long as they're not *too* much longer). The next lemma is in the same spirit.

Lemma 4.7.6. *Let G be a d-regular graph, $d \geq 7$, with n vertices and girth g, and consider an L'-step random walk, with $g/2 \leq L' \leq kg/2$, for some integer $k \geq 2$, that begins at a fixed vertex v. Then the probability that*

(i) in the first $g/2$ steps the walk leaves the g/d-ball around v, and

(ii) the walk never returns to this g/d-ball in subsequent steps

is at least $1-6ke^{-\Omega(g)}$. In particular, if $g=\Omega(\log_{d-1} n)$ then the probability that (i) and (ii) both hold is at least $1-6kn^{-\Omega(1)/\log(d-1)}$.

Proof. We will prove that the probability that (i) fails to hold is at most $e^{-\Omega(g)}$ and the probability that (ii) fails to hold is at most $6(k-1)e^{-\Omega(g)}$, at which point the union bound will prove the claim (the special case of logarithmic girth is self-evident).

Lemma 4.7.2 tells us that the probability of the first $g/2$ steps ending less than $g/6$ away from v is $e^{-\Omega(g)}$. Moreover, it is easy to see that the $L/3$ in that lemma is somewhat arbitrary, and $1/3$ can be replaced by any constant strictly less that $\frac{d-1}{d} \geq \frac{6}{7}$. In particular, if we replace $1/3$ by $1/3 + 2/d$ then we find that the probability that a $g/2$-step walk that starts at v ends at distance less than $(1/6 + 1/d)g$ from v is $e^{-\Omega(g)}$.

To rephrase, the probability that after the first $g/2$ steps the walk is at distance less than $g/6$ from the g/d-ball around v is at most $e^{-\Omega(g)}$. This proves the claim about (i). As for (ii), it now suffices to prove that in the remainder of the walk, which consists of at most $\frac{g}{2}(k-1)$ steps, the probability that we walk more than $g/6$ steps towards v is $6(k-1)e^{-\Omega(g)}$, for this will establish that (ii) fails to hold with probability at most $6(k-1)e^{-\Omega(g)}$. We will in fact prove the stronger claim that the probability of taking more than $g/12$ steps towards v is so bounded.

Note that the remainder of the walk can take us outside of the $g/2$-ball around v. We are unable to reason about the walk when it is outside this $g/2$-ball, but fortunately we don't have to; walking outside the $g/2$-ball doesn't bring the walk any closer to v. Thus in the remainder of the proof we will condition on the walk never leaving the the $g/2$-ball around v; once we prove our probability bound in this worst-case situation, we can safely extend it to arbitrary walks that leave and enter the $g/2$-ball around v at will.

In light of all this, it suffices to show that for a random walk of $\frac{g}{12}$ steps that begins at some vertex inside the $g/2$-ball around v that is fairly far from the g/d-ball around v, the probability that this walk takes at least $\frac{g}{72(k-1)}$ steps towards v is at most $e^{-\Omega(g)}$. Once we show this, it will follow that in the remainder of our original walk, which consists of at most $\frac{g}{2}(k-1)$ steps, the probability that we take at least $g/12$ steps towards v is at most $6(k-1)e^{-\Omega(g)}$; simply break up this walk into at most $6(k-1)$ walks of at most $g/12$ steps and use the union bound.

So let us consider a $g/12$-step random walk that begins at some u inside the $g/2$-ball around v, where u is at least $g/12 + g/d$ away from v. What is the probability that this walk takes at least $\frac{g}{72(k-1)}$ steps towards v? As in the proofs of Lemmas 4.7.2 and 4.7.4, we observe that since the $g/2$-ball around v is a d-regular tree rooted at v, each step inside this ball is decided by the flip of a coin whose probability of heads is $p = \frac{d-1}{d}$; heads means move away from v while tails means move towards v. If the walk takes at least $\frac{g}{72(k-1)}$ steps towards v then the number of coin flips that came up heads is at most

$$\frac{g}{12}\left(\frac{1}{2} + \frac{1}{12(k-1)}\right) \leq \left(\frac{7}{12}\right)\frac{g}{12}.$$

By the Chernoff bound (Lemma 4.7.1), the probability of this event is $e^{-\Omega(g)}$, as desired. □

4.7.2 Proof overflow

In this section we supply proofs for two technical results that were used in the proof of Theorem 4.4.1: Lemma 4.4.3, and claim (v) in the proof of Theorem 4.4.2. We begin with the former.

In Section 4.4 we stated Lemma 4.4.3 without proof and used it to prove Theorem 4.4.2, from which our main result Theorem 4.4.1 followed (as discussed in Section **??**). Before proving the lemma, recall its statement: we have a d-regular graph G on n vertices such that $|\lambda_i(G)| \leq \eta$ for $i \geq 2$, say $\eta = .05$. We fix a master tour of $V(G)$, some integer k, and set $\ell = n/(32k)$. We claim that if w is a uniformly sampled k-step random walk on G, then the probability that w encounters a vertex whose index is in $[w(1) : \ell]_n$ is less than $1/4$ (recall that $w(1)$ is the index of the first vertex encountered by w).

Proof of Lemma 4.4.3. Formalizing the claim, we would like to show that

$$\Pr\left[\exists t \in \{1, \ldots, k\} \text{ such that } w(t) \in [w(1) : \ell]_n\right] < \frac{1}{4}$$

Let us use the notation w_j for a k-step random walk that is conditioned to begin at some fixed index j, i.e. a walk such that $w_j(1) = j$. Then, since w is a uniformly sampled k-step random walk, w is w_j for a j that is chosen uniformly. Therefore,

$$\Pr\left[\exists t \in \{1, \ldots, k\} \text{ such that } w(t) \in [w(1) : \ell]_n\right] = \frac{1}{n}\sum_{j=1}^{n} \Pr\left[\exists t \in \{1, \ldots, k\} \text{ such that } w_j(t) \in [j : \ell]_n\right]$$

and by the union bound we have

$$\Pr\left[\exists t \in \{1, \ldots, k\} \text{ such that } w(t) \in [w(1) : \ell]_n\right] \leq \frac{1}{n}\sum_{j=1}^{n}\sum_{t=1}^{k} \Pr\left[w_j(t) \in [j : \ell]_n\right].$$

Thus it suffices to show that

$$\sum_{j=1}^{n}\sum_{t=1}^{k} \Pr\left[w_j(t) \in [j : \ell]_n\right] < \frac{n}{4}. \tag{4.8}$$

To this end, we make a few definitions. Given $j \in [n]$, let the vector $\mathbf{u}_j$ be the characteristic vector for the set $[j : \ell]_n$ and let $\mathbf{e}_j$ be the jth standard basis vector.

Let M be the transition matrix for the random walk, i.e. $M = \frac{1}{d}A$ with A the adjacency matrix of our graph G. Since we can write

$$\Pr\left[w_j(t) \in [j:\ell]_n\right] = \mathbf{u}_j^\top M^t \mathbf{e}_j$$

it follows that

$$\sum_{j=1}^{n}\sum_{t=1}^{k} \Pr\left[w_j(t) \in [j:\ell]_n\right] = \sum_{j=1}^{n} \mathbf{u}_j^\top \left(\sum_{t=1}^{k} M^t\right) \mathbf{e}_j.$$

Using some matrix trace trickery,

$$\begin{aligned}\sum_{j=1}^{n} \mathbf{u}_j^\top \left(\sum_{t=1}^{k} M^t\right) \mathbf{e}_j &= \operatorname{tr}\left[\sum_{j=1}^{n} \mathbf{u}_j^\top \left(\sum_{t=1}^{k} M^t\right) \mathbf{e}_j\right] \\ &= \operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right)\left(\sum_{j=1}^{n} \mathbf{e}_j \mathbf{u}_j^\top\right)\right].\end{aligned}$$

For notational convenience let us define the matrix

$$N_{\ell,n} = \sum_{j=1}^{n} \mathbf{e}_j \mathbf{u}_j^\top.$$

By definition, N is a 0-1 matrix whose j, j' entry is 1 if and only if $j' \in [j:\ell]_n$. For example,

$$N_{2,6} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Employing our new shorthand, we have

$$\sum_{j=1}^{n}\sum_{t=1}^{k} \Pr\left[w_j(t) \in [j:\ell]_n\right] = \operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right) N_{\ell,n}\right] \tag{4.9}$$

so that, recalling (4.8), our aim becomes to prove that

$$\operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right) N_{\ell,n}\right] < \frac{n}{4}. \tag{4.10}$$

To this end, another definition. For $s = 0, 1, \ldots, n/\ell - 1$ we define $\mathbf{v}_s$ to be the characteristic vector of the set $\{s\ell + 1, s\ell + 2, \ldots, (s+2)\ell\}$. In words, $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$, etc. are, respectively, the characteristic vectors of the index sets 1 to 2ℓ, $\ell + 1$ to 3ℓ, $2\ell + 1$ to 3ℓ, etc. We wraparound for $s = n/\ell - 1$, so that $\mathbf{v}_{n/\ell-1}$ is the characteristic vector of the set $\{n - \ell + 1, \ldots, n\} \cup \{1, \ldots, \ell\}$. Observe that $\mathbf{v}_s \mathbf{v}_s^\top$ is a matrix consisting of a $2\ell \times 2\ell$ block of ones (for $s = n/\ell - 1$ this block is broken up into four $\ell \times \ell$ blocks of ones at the corners).

Consider the sum of the $\mathbf{v}_s \mathbf{v}_s^\top$; for example, if $n = 6$ and $\ell = 2$ then

$$\sum_{s=0}^{n/\ell-1} \mathbf{v}_s \mathbf{v}_s^\top = \begin{pmatrix} 2 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 \end{pmatrix}$$

though in general this matrix will also contain zeros. It is easy to see that every entry of $\sum_s \mathbf{v}_s \mathbf{v}_s^\top$ is greater than or equal to the corresponding entry of $N_{\ell,n}$.

Since $\sum_s \mathbf{v}_s \mathbf{v}_s^\top$, $N_{\ell,n}$, and $\sum_t M^t$ all have non-negative entries, it follows that

$$\begin{aligned}
\operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right) N_{\ell,n}\right] &\le \operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right)\left(\sum_{s=0}^{n/\ell-1} \mathbf{v}_s \mathbf{v}_s^\top\right)\right] \\
&= \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \operatorname{tr}\left[M^t \mathbf{v}_s \mathbf{v}_s^\top\right] \\
&= \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \operatorname{tr}\left[\mathbf{v}_s^\top M^t \mathbf{v}_s\right] \\
&= \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \mathbf{v}_s^\top M^t \mathbf{v}_s.
\end{aligned}$$

Let us write $\mathbf{v}_s = \frac{2\ell}{n}\mathbf{1} + \mathbf{v}_s^\perp$ where $\mathbf{1}$ is the all-ones vector and $\mathbf{v}_s^\perp$ is orthogonal to $\mathbf{1}$. Now we have

$$\begin{aligned}
\operatorname{tr}\left[\left(\sum_{t=1}^{k} M^t\right) N_{\ell,n}\right] &\le \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \mathbf{v}_s^\top M^t \mathbf{v}_s \\
&= \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \left(\frac{2\ell}{n}\mathbf{1} + \mathbf{v}_s^\perp\right)^\top M^t \left(\frac{2\ell}{n}\mathbf{1} + \mathbf{v}_s^\perp\right) \\
&= \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \left(\frac{2\ell}{n}\right)^2 \mathbf{1}^\top M^t \mathbf{1} + \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \left(\mathbf{v}_s^\perp\right)^\top M^t \mathbf{v}_s^\perp \qquad (4.11)
\end{aligned}$$

where the last line follows from the fact that $\mathbf{1}$ is an eigenvector of M and is orthogonal to $\mathbf{v}_s^\perp$.

Since $\mathbf{1}$ is indeed an eigenvector of M with eigenvalue 1, we have

$$\sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \left(\frac{2\ell}{n}\right)^2 \mathbf{1}^\top M^t \mathbf{1} = \sum_{t=1}^{k} \sum_{s=0}^{n/\ell-1} \left(\frac{2\ell}{n}\right)^2 n = \frac{4\ell^2}{n} \cdot \frac{n}{\ell} k = 4\ell k = \frac{n}{8} \qquad (4.12)$$

since we've defined $\ell = n/(32k)$. As for the second summation, recall that if $\lambda_2(M^t)$ is the second-largest eigenvalue of M_t then Rayleigh's inequality tells us that

$$\frac{\left(\mathbf{v}_s^\perp\right)^\top M^t \mathbf{v}_s^\perp}{\left(\mathbf{v}_s^\perp\right)^\top \mathbf{v}_s^\perp} \le \lambda_2(M^t)$$

for every $\mathbf{v}$ orthogonal to $\mathbf{1}$ (since $\mathbf{1}$ is the eigenvector for the largest eigenvalue). Recall that we have a bound of η on the non-principal eigenvalues of G, so the second-largest eigenvalue of M^t is bounded by η^t. It follows that

$$\sum_{t=1}^{k}\sum_{s=0}^{n/\ell-1}\left(\mathbf{v}_s^\perp\right)^\top M^t\mathbf{v}_s^\perp \le \sum_{t=1}^{k}\sum_{s=0}^{n/\ell-1}\eta^t\left(\mathbf{v}_s^\perp\right)^\top \mathbf{v}_s^\perp = \left(\sum_{t=1}^{k}\eta^t\right)\left(\sum_{s=0}^{n/\ell-1}\left(\mathbf{v}_s^\perp\right)^\top \mathbf{v}_s^\perp\right).$$

Now,

$$\left(\mathbf{v}_s^\perp\right)^\top \mathbf{v}_s^\perp = \left(\mathbf{v}_s - \frac{2\ell}{n}\mathbf{1}\right)^\top\left(\mathbf{v}_s - \frac{2\ell}{n}\mathbf{1}\right) = \mathbf{v}_s^\top\mathbf{v}_s - \frac{2\ell}{n}\mathbf{v}_s^\top\mathbf{1} + \left(\frac{2\ell}{n}\right)^2\mathbf{1}^\top\mathbf{1} = \mathbf{v}_s^\top\mathbf{v}_s = 2\ell$$

and so

$$\sum_{t=1}^{k}\sum_{s=0}^{n/\ell-1}\left(\mathbf{v}_s^\perp\right)^\top M^t\mathbf{v}_s^\perp \le \left(\sum_{t=1}^{k}\eta^t\right)\left(\sum_{s=0}^{n/\ell-1}\left(\mathbf{v}_s^\perp\right)^\top \mathbf{v}_s^\perp\right) \le \frac{\eta}{1-\eta}\sum_{s=0}^{n/\ell-1}2\ell \le \frac{\eta}{1-\eta}2n. \tag{4.13}$$

Putting inequalities (4.11), (4.12) and (4.13) together, we have

$$\mathrm{tr}\left[\left(\sum_{t=1}^{k}M^t\right)N_{\ell,n}\right] \le \left(\frac{1}{8}+\frac{2\eta}{1-\eta}\right)n.$$

We assumed that $\eta = .05$, hence $2\eta(1-\eta)^{-1} < 1/8$. This proves equation (4.10), hence equation (4.8), and completes the proof. □

The second technical result left to prove is a claim in the proof of Theorem 4.4.2 in Section 4.4 (specifically, the claim that the event (v) described in that proof happens with probability at least $4/5$). Let us rephrase this claim as a lemma.

Lemma 4.7.7. *Let $G = (V, E)$ be a d-regular Ramanujan graph with $|V| = n$ and girth $g \ge \frac{2}{3}\log_{d-1} n$, and consider some $U \subseteq V$ with $|U| = n/(16g)$. Let w be a random walk of $k \ge 139g/2$ steps that begins from an arbitrary distribution on V. For sufficiently* *large d there exists n_0 such that if $n \geq n_0$, then the probability that w encounters a vertex in U in its last $64g$ steps is at least $4/5$.*

Proof. As before, let M be the transition matrix for the random walk on G and let its eigenvalues be $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Since G is Ramanujan, we know that $|\lambda_i| \leq \frac{2}{d}\sqrt{d-1}$ for $i \geq 2$. Let $\mathbf{p}$ be the distribution of the walk over V after the first $k - 64g \geq 11g/2$ steps. Finally, let D be a 0-1 diagonal matrix such that $D_{ii} = 1$ iff the ith vertex is *not* in U.

Recall that $M^t\mathbf{p}$ is the distribution of step $k - 64g + t$ of w. It follows that $||DM^t\mathbf{p}||_1$ is the probability that step $k - 64g + t$ of w is not in U. We claim that the probability of w avoiding U in its last $64g$ steps is at most $1/5$. It is easy to check this this is equivalent to

$$||(DM)^{64g}\mathbf{p}||_1 \leq \frac{1}{5}. \tag{4.14}$$

It is well known (e.g. Section 3.1 of [28]) that on a regular graph G such that $|\lambda_i(G)| \leq \eta$ for $i \geq 2$, if a random walk is begun from an arbitrary distribution over V and we write $p_i^{(s)}$ for the probability that the sth vertex encountered by the walk is i, then $|p_i^{(s)} - 1/n| \leq \eta^s\sqrt{n}$. Setting $s = k - 64g \geq 11g/2$ and recalling the Ramanujan condition (as well as the lower bound on g) gives

$$|\mathbf{p}_i - 1/n| \leq \eta^s\sqrt{n} \leq \eta^{11g/2}\sqrt{n} \leq \left(2(d-1)^{-1/2}\right)^{\frac{11}{3}\log_{d-1} n}\sqrt{n} \leq n^{\frac{1}{2} + \frac{11}{3\log(d-1)} - \frac{11}{6}}.$$

If we take d sufficiently large then $|\mathbf{p}_i - 1/n| \leq n^{-5/4}$. Therefore if we write $\boldsymbol{\pi}$ for the uniform distribution over V we find that $\mathbf{p} = \boldsymbol{\pi} + \boldsymbol{\delta}$ with $|\boldsymbol{\delta}_i| \leq n^{-5/4}$ for each i.

Returning to inequality (4.14), begin by writing $||(DM)^{64g}\mathbf{p}||_1 \leq ||(DM)^{64g}\boldsymbol{\pi}||_1 + ||(DM)^{64g}\boldsymbol{\delta}||_1$. It is easy to see that if $\mathbf{u}$ is a non-negative vector then $||DM\mathbf{u}||_1 \leq ||\mathbf{u}||_1$, and this implies

$$||(DM)^{64g}\boldsymbol{\delta}||_1 \leq ||\boldsymbol{\delta}||_1 \leq n^{-1/4}.$$

It remains to bound $||(DM)^{64g}\boldsymbol{\pi}||_1$. Observe that this quantity is precisely the probability that a uniformly sampled random walk of $64g$ steps is contained entirely in $V \setminus U$. We can bound this probability using the following result, which is Lemma 1 from [21]:

Lemma 4.7.8 (Goldreich et al. [21])**.** *Let $G = (V, E)$ be a regular graph such that $|\lambda_i(G)| \leq 1/2$ for $i \geq 2$, fix some $W \subseteq V$, and set $\mu = |W|/|V|$. If $\mu \geq 1/2$ then the probability that a uniformly sampled k-step random walk is contained in W is at most $\mu^{k/2}$.*

Applying this lemma to $W = V \setminus U$ with $k = 64g$ gives

$$||(DM)^{64g}\boldsymbol{\pi}||_1 \leq \left(1 - \frac{1}{16g}\right)^{32g} \leq e^{-2}.$$

Putting it all together, we have

$$||(DM)^{64g}\mathbf{p}||_1 \leq ||(DM)^{64g}\boldsymbol{\pi}||_1 + ||(DM)^{64g}\boldsymbol{\delta}||_1 \leq e^{-2} + n^{-1/4}.$$

When n is sufficiently large, the right-most side is less than $1/5$, which establishes inequality (4.14) and completes the proof. □

BIBLIOGRAPHY

[1] A.A. Ageev and M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[2] R. Andersen and U. Feige. Interchanging distance and capacity in probabilistic mappings. *http://arxiv.org/abs/0907.3631*, 2009.

[3] E. Anshelevich, D. Chakrabarty, A. Hate, and C. Swamy. Approximation algorithms for the firefighter problem: Cuts over time and submodularity. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISAAC)*, pages 974–983, 2009.

[4] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[5] J. Bartholdi and L. Platzman. An $o(n \log n)$ planar traveling salesman heuristic based on spacefilling curves. *Operations Research Letters*, 1(4):121–125, 1981.

[6] J. Bartholdi, L. Platzman, R. L. Collins, and W. H. Warden III:. A minimal technology routing system for meals on wheels. *Interfaces*, 13(3):1–8, 1983.

[7] D. Bertsimas and M. Grigni. On the spacefilling curve heuristic for the euclidean traveling salesman problem. *Operations Research Letters*, 8(5):241–244, 1989.

[8] D. Bertsimas, P. Jaillet, and A. Odoni. On the spacefilling curve heuristic for the euclidean traveling salesman problem. *Operations Research Letters*, 38(6):1019–1033, 1990.

[9] B. Bollobás. *Modern Graph Theory*. Springer-Verlag, 2001.

[10] L. Cai, E. Verbin, and L. Yang. Firefighting on trees: $(1 - 1/e)$-approximation, fixed parameter tractability and a subexponential algorithm. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 258–269, 2008.

[11] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. In *Proceedings of the Symposium on Theory of Computing (STOC)*, 2008.

[12] H. Chernoff. A note on an inequality involving the normal distribution. *Annals of Probability*, 9(3):533–535, 1981.

[13] Colorado.Dept.Agriculture. Colorado biocontrol. *Published online at www.colorado.gov*, Spring Issue, 2009.

[14] G. Cumming. Habitat shape, species invasions, and reserve design: Insights from simple models. *Conservation Ecology*, 6 (1), 2002.

[15] R. Engelberg, J. Könemann, S. Leonardi, and J. Naor. Cut problems in graphs with a budget constraint. In *7th LATIN*, pages 435–446, 2006.

[16] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computational Syst. Science*, 69(3):485–497, 2004.

[17] M.A. Finney. Fire growth using minimum time travel methods. *Canadian Forest Resources*, 32:1420–1424, 2002.

[18] M.A. Finney. A computational method for optimising fuel treatment locations. *International Journal of Wildland Fire*, 16:702–711, 2007.

[19] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming Study*, 8:73–87, 1978.

[20] K.M. Gebert and E.G. Schuster. Forest service fire suppression expenditures in the southwest1. *USDA Forest Service General Technical Report*, PSW-GTR-189, 2008.

[21] O. Goldreich, R. Impagliazzo, L.A. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *Founadtions of Computer Science (FOCS)*, pages 318–326, 1990.

[22] I. Gorodezky, R.D. Kleinberg, D.B. Shmoys, and G. Spencer. Improved lower bounds for the universal and *a priori* tsp. In *Proceedings of APPROX-RANDOM*, pages 178–191, 2010.

[23] A. Gupta, M.T. Hajiaghayi, and H. Räcke. Oblivious network design. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 970–979, 2006.

[24] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 417–426, 2004.

[25] M.T. Hajiaghayi, R.D. Kleinberg, and F.T. Leighton. Improved lower and upper bounds for universal tsp in planar metrics. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 649–658, 2006.

[26] B. Hartnell. Firefighter! an application of domination. *Presentation, 25th Manitoba Conference of Combinatorial Mathematics and Computing*, 1995.

[27] A. Hayrapetyan, D. Kempe, M. Pál, and Z. Svitkina. Unbalanced graph cuts. In *13th European Symposium on Algorithms (ESA)*, pages 191–202, 2005.

[28] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43:439–561, 2006.

[29] P. Jaillet. Probabilistic traveling salesman problems. *Technical Report, MIT Operations Research Center*, 185, 1985.

[30] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for tsp, steiner tree, and set cover. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 386–395, 2005.

[31] D. Kempe, J.M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[32] K. Kovacs, R.G. Haight, D.G. McCullough, R.J. Mercader, N.W. Siegert, and A.M. Liebhold. Cost of potential emerald ash borer damage in u.s communities, 2009-2019. *Ecological Economics*, 69:569–578, 2010.

[33] K.F. Kovacs, R.G. Haight, and R.J. Mercader. A bioeconomic analysis of emerald ash borer control in an urban forest. *Working Paper*, 2012.

[34] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[35] E.S. Minor and R.H. Gardner. Landscape connectivity and seed dispersal characteristics inform the best management strategy for exotic plants. *Ecological Applications*, 21(3):739–749, 2011.

[36] M. Morgenstern. Existence and explicit constructions of + 1 regular ramanujan graphs for every prime power . *Journal of Combinatorial Theory, Ser. B*, 62(1):44–62, 1994.

[37] A. Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991.

[38] R.J. Norris, J. Memmot, and D.J. Lovell. The effect of rainfall on the survivorship and establishment of a biological control agent. *Journal of Applied Ecology*, 39:226–234, 2002.

[39] L.K. Platzman and J.J. Bartholdi III. Spacefilling curves and the planar travelling salesman problem. *Joural of ACM*, 36(4):719–737, 1989.

[40] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *40th Symposium on Theory of Computing (STOC)*, pages 255–264, 2008.

[41] S. M. Ross. *Introduction to Probability Models*. Academic Press, 2007.

[42] F. Schalekamp and D.B. Shmoys. Algorithms for the universal and *a priori* tsp. *Operations Research Letters*, 36(1):1–3, 2008.

[43] K. Shea and H.P. Possingham. Optimal release strategies for biological control agents: an application of stochastic dynamic programming to population management. *Journal of Applied Ecology*, 37:77–86, 2000.

[44] D.B. Shmoys and G. Spencer. Approximation algorithms for fragmenting a graph against a stochastically-located threat. In *Proceedings of the Workshop on Approximation and Online Algorithms (WAOA)*, 2011.

[45] D.B. Shmoys and K. Talwar. A constant approximation algorithm for the *a priori* traveling salesman problem. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 331–343, 2008.

[46] G. Spencer. Exposition on tree-to-graph extensions for bud-

geted objectives linear in pairwise disconnection. *Online at http://people.orie.cornell.edu/gms39/ images/documents/spencerttgexpos.pdf*.

[47] G. Spencer. Robust cuts over time: Combatting the spread of invasive species with unreliable biological control. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI 2012)*, 2012.

[48] M.I. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43, 2004.

[49] Y. Wei, D. Rideout, and A. Kirsch. An optimization model for locating fuel treatments across a landscape to reduce expected fire losses. *Canadian Journal of Forest Resources*, 38:868–877, 2008.