OPTIMIZATION AND SIMULATION MODELS FOR THE DESIGN OF RESILIENT ELECTION VOTING SYSTEMS

by
Adam P. Schmidt

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Industrial and Systems Engineering)

at the
UNIVERSITY OF WISCONSIN–MADISON
2022

Date of final oral examination: 06/07/2022

The dissertation is approved by the following members of the Final Oral Committee:
Laura A. Albert, Professor, Industrial and Systems Engineering
Oguzhan Alagoz, Professor, Industrial and Systems Engineering
Justin J. Boutilier, Assistant Professor, Industrial and Systems Engineering
Barry C. Burden, Professor, Political Science
Duncan A. Buell, Professor Emeritus, Computer Science and Engineering (University of South Carolina)

© Copyright by Adam P. Schmidt 2022
All Rights Reserved

To my family.

Acknowledgments

First and foremost, I want to thank my advisor, Professor Laura Albert, whose expertise was invaluable to my growth as a researcher and to the construction of this dissertation. My gratitude extends to Professor Oguzhan Alagoz, Professor Justin J. Boutilier, Professor Emeritus Duncan A. Buell, and Professor Barry Burden, the members of my dissertation committee and the collaborators of my research, who have provided valuable insight and guidance that has strengthened the ability of my research to have a lasting impact on our society. I also want to thank Professor Jeffrey Linderoth who is the reason I became interested in operations research and has provided me valuable guidance during my time at UW-Madison. I would also like to thank Dr. Kay (Kaiyue) Zheng who was my mentor during my first research experience as an undergraduate student.

I would like to thank the Richland County Government and Milwaukee Election Commission, specifically Alexandria Stephens and Claire Woodall-Vogg, for providing insight into the challenges related to election planning.

I want to acknowledge my fellow PhD students who have provided support and guidance along the way. Particularly, I want to thank Veronica White, Rachel Zenker, Ahkilesh Soni, and Eric Stratman for sharing the PhD process with me. Lastly, I would like to thank my family and friends, although some are no longer here to see this accomplishment. To my parents, Eric and Angela, and my siblings, Zach and Katie, for their love and support my entire life. To my dear friends Allison, Colin, Courtney, Emma, Lauren, and Nick. Without

their support, I would not have completed this degree nor be the person I am today.

This dissertation is based in part on the previously published articles and manuscripts in preparation listed below. I have permission from my co-authors/publishers to use the works listed below in my dissertation.

- Adam P. Schmidt, & Laura A. Albert (2022). Designing pandemic-resilient voting systems. *Socio-Economic Planning Sciences*, 80, 101174.
- Adam P. Schmidt, & Laura A. Albert (2022). Task recommendations for self-assigning spontaneous volunteers. *Computers & Industrial Engineering*, 163, 107798.
- Carmen Haseltine, Adam Schmidt, and Laura A. Albert. (2022). On designing and operating Vote-by-Mail processes. *Technical Report*.
- Adam P. Schmidt, Duncan Buell, & Laura A. Albert (2022+). Optimal polling location consolidation. *Manuscript in preparation*.
- Adam P. Schmidt, & Laura A. Albert (2022+). An optimization model for locating ballot drop boxes. *Manuscript in preparation*.

Contents

Abstract

Election voting systems are defined by complex inter-related processes, and design decisions made by election officials have been shown to influence voter participation. The elections held during the COVID-19 pandemic demonstrated the vulnerability of our voting systems to changing election conditions. Unforeseen disruptions to a voting system require election officials to quickly respond to changing election conditions and redesign the voting system. This motivates the need for analytical tools to help election officials design systems that perform well under a range of election conditions and to help develop contingency plans to use in response to an election disruption.

This dissertation introduces new optimization and simulation models to support the design of resilient voting systems. Using these models, we identify analytically founded election practices that make election voting systems safer, more reliable, and more equitable. We first study the impact of pandemic-related disruptions on in-person voting processes and study practices to mitigate these disruptions using a discrete event simulation of the in-person voting process. We then study when and how to consolidate polling locations given the constraints and criteria election officials should follow. We formulate an integer program of the polling location consolidation problem, study its theoretical properties, and study implications for practice using a case study of Richland County, South Carolina. We then study how to select ballot drop box locations that balance the trade-off between cost, risk, and equitable accessibility of the voting system. We introduce an integer program

of the drop box location problem and introduce a heuristic to generate a large number of feasible solutions for policy makers to select from a posteriori. We quantify the benefit of using optimization for this problem using a case study of Milwaukee, WI. Lastly, changes to voting systems and processes may incline some voters not to participate in an election. The actions taken by volunteers can increase voter participation. We study how to indirectly manage self-assigning spontaneous volunteers using tailored task recommendations. We introduce an integer program to construct recommendations for volunteers and highlight the benefit of the task recommendations using a real-world case study of disaster response efforts.

Chapter 1

Introduction

1.1 Motivation

The election voting infrastructure within the United States (U.S.) is critical to the functioning of our democracy, and as a result, elections-related voting systems, polling locations, and IT infrastructure are classified as components of our *critical infrastructure* [53]. Critical infrastructure are "systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters" [1].

Not only is the voting infrastructure necessary for the functioning of our democracy, it has a lasting impact on our nation. Elections are solely determined by the individuals that chose to participate in an election, but the voting infrastructure and voting processes, in part, determine who ultimately casts a ballot. Research suggests that voters determine whether to participate by comparing the "cost" of voting against the benefit they would receive from voting. The cost of voting is influenced by a number of factors, including the voting infrastructure. Two of the most influential factors are the distance that voters

must travel to cast a ballot and the amount of time it takes to complete the voting process, particularly for those with a low socioeconomic status. These costs are in part determined by the location and allocation of critical voting resources during an election.

The elections held during the COVID-19 pandemic demonstrated the vulnerability of our voting systems to changing election conditions. Election administrators typically spend months preparing for an election and implementing strategies to ensure that voting is effective, equitable, accessible, and quick. However, during the COVID-19 pandemic, the elections occurred while many states operated under changing emergency orders related to the pandemic, which placed unusual strains on the voting system. Election administrators were forced to deviate from previously used practices and design modified voting systems. In many cases, voters ultimately faced long wait times at their polling location and had to travel further to vote than in previous elections.

The elections also highlighted the susceptibility of voting systems to changing voter behavior. During the 2020 general election, a record 46% of voters cast their ballot using the absentee voting system [131], and the volume of ballots revealed new problems for election officials to consider. Operating the absentee voting system at scale substantially increased the cost of facilitating an election and increased the risk of malicious and non-malicious attacks on the absentee voting process.

While the COVID-19 pandemic illuminated the challenges of designing a voting system to the general public, many challenges were prevalent prior to the pandemic and will continue to be challenges in the future. Voting systems are designed by a small number of election administrators with limited time, resources, and budget which necessitates trade-offs between election related goals. However, voting systems are defined by complex, inter-related processes with multiple stakeholders who typically have conflicting objectives. Reducing the voter-experienced "cost" of voting often increases the financial cost of the voting system, and under some election conditions, reducing some "costs" of voting may

increase others. Moreover, elections are a one-time event with no possible recourse action after the election is completed, and many of the decisions must be made before the actual conditions on Election Day are known. These decisions are becoming even more challenging, since how voters interact with the voting system is changing. As illustrated in Figure 1.1, the proportion of voters using non-traditional voting methods (absentee or in-person prior to Election Day) has steadily grown since 1996 and reached 69.4% of voters in 2020 [165]. The way election officials adapt to changing voter behavior can result in inequitable voting systems and raise questions of disenfranchisement of minority populations [151].

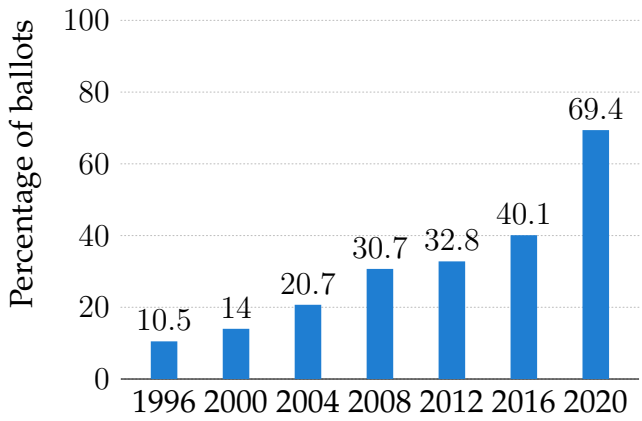


Figure 1.1: The percentage of ballots cast using non-traditional (absentee or early in-person) voting methods [165].

The challenge and importance of planning voting systems that are resilient against changing election conditions motivate the need for analytical models of voting systems that can be used by election administrators to plan and assess their voting systems, particularly when they cannot rely on past experiences due to disruptions, disasters, or security incidents. Analytical models provide a structured and rigorous approach to making complex, interrelated decisions and can serve as the technical basis for practices that can be used by election officials to design voting systems that improve voter participation, reduce financial cost, and mitigate voting system risks.

1.2 Background and Related Literature

In the U.S., states are divided into *legislative districts* that collectively elect a member to a specific legislative body. By law, legislative districts must be constructed to ensure equal representation, commonly referred to as "one person, one vote." Legislative districts are typically large and are divided into *voting precincts* during an election. These smaller, often contiguous voter precincts represent residents that vote at the same location.

Recently, there has been increased interest among the operations research community in (re)designing legislative districts to achieve "one person, one vote." The first optimization-based approach for the political redistricting problem was proposed by Hess [91], and many authors have recently proposed alternative algorithmic or optimization-based approaches to achieve this goal [57, 111, 70]. However, even perfectly designed congressional districts do not ensure equal representation among the individuals that actually cast a ballot. In the U.S., voting is not compulsory, and individuals choose whether to participate in an election. According to an economic theory of election participation, potential voters decide whether to vote by comparing the "cost" of voting and the potential benefits from voting [56]. The cost of voting is influenced by a number of factors, including the weather [74] and availability of political information [56]. More importantly for election officials, the design of the voting system infrastructure and the allocation of voting resources in part determine the cost of voting [170, 88]. Long wait times, long voting queues, and long distances to polling locations each substantially impact the voting experience, reduce the trust voters have in the election, and reduce the fraction of eligible voters that cast a ballot, known as *voter turnout* [26, 61, 188, 80, 88, 92]. Thus, the design of the voting system processes and infrastructure deserves special attention when considering the goal of "one person, one vote."

The rules and regulations pertaining to election administration and the election voting

infrastructure differ by county or municipality. Throughout the remainder of the dissertation, I will refer to a "typical" voting process that describes the voting system experienced by a large proportion of voters across the country. Not all jurisdictions follow this same process, but the differences are generally small. Within each jurisdiction, voting systems are managed by election administrators or officials. In the typical voting system, voter precincts are assigned to a single polling location at which individuals can vote. Each polling location may have one or more voter precincts assigned to it, and there are typically multiple polling locations placed throughout the jurisdiction. Each polling location is assigned multiple poll workers who manage the voting processes and help voters cast a ballot. The polling locations are also allocated a number of voting resources (e.g., voting machines, optical ballot scanners) that are necessary for the voting process. Voters can cast a ballot at a polling location either on Election Day or leading up to Election Day, known as the early voting period. This process is referred to as the *in-person* voting system. In many jurisdictions, there is a second method by which voters can cast a ballot, referred to as the *absentee* voting system[1]. In this voting pathway, a voter can request a ballot be mailed to their residence. The completed ballot is either returned by mail or placed in a ballot drop box if allowed by law. In some states, voters must provide a reason to vote absentee, while "no-excuse" absentee voting is allowed in 34 states [140]. The in-person and absentee voting pathways are summarized in Figure 1.2.

There are many inter-related decisions that election administrators must make regarding the design of the voting system. First, election administrators must select polling locations at which voters can vote in-person. Election administrators must also determine how voter precincts should be assigned to polling locations in a way that minimizes the distance voters must travel to their polling location. Voting resources, such as poll workers and

[1]Some states, such as Washington, use the "absentee" voting process as their primary voting method. Thus, I use "absentee" loosely in this dissertation and sometimes refer to it as the vote-by-mail process.

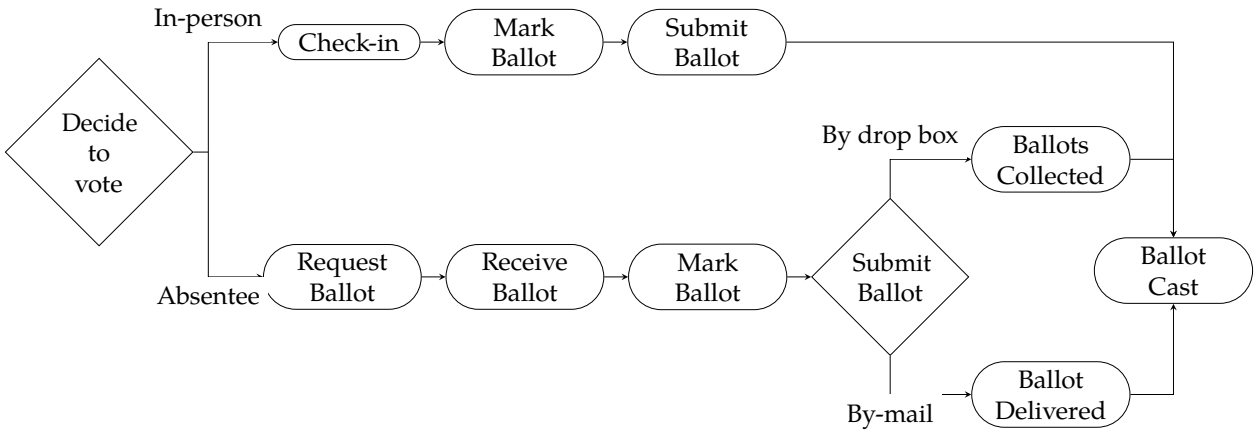


Figure 1.2: Typical pathways to cast a ballot, divided into in-person and absentee.

voting machines, must be allocated to the polling locations to minimize the wait time experienced by the voters. The layout of each polling location must then be determined to ensure a smooth flow of voters. Equally challenging are the decisions related to the absentee voting system. Election administrators must determine the location of ballot drop boxes and the allocation of staff for processing absentee ballots. In both systems, election administrators must implement practices to mitigate the risk of malicious and non-malicious attacks on the voting system. The goal is to design voting systems in which the voter-experienced cost of voting is minimized so that voter turnout is maximized. However, election administrators typically have a fixed budget and a fixed number of resources at their disposal. This necessitates trade-offs when making these decisions. A challenge to election planning is ensuring the voting system is robust to a number of election conditions and understanding how voters will respond to potential voting system designs. The voters' decision to participate (or not) and the voting pathway they decide to utilize ultimately determines the cost-effectiveness, equity, and performance of the voting system.

Considering the complexity and importance of the decisions made by election administrators, the study of U.S. voting system design within the operations research community is relatively limited. Some authors have attempted to identify and describe the operation and

performance of in-person voting processes [181, 186]. Others applied queueing theory to analyze lines at polling locations and identify mitigating practices to avoid long lines [190]. Voting machines have been recognized as a bottleneck in the in-person voting process [221], and as a result, a stream of papers has focused on the allocation of voting machines to polling locations [6, 214, 65]. Other research attempted to determine, after an election occurred, if voting system design resulted in voter disenfranchisement. There have been studies into the allocation of voting machines to polling locations [92, 6], the cause of long wait times [21], and the accuracy of voting machine tallies [20]. In general, the research is restricted to the operation and processes of in-person voting. Four papers study the selection of polling locations, but no paper studies the U.S. voting system [79, 138, 60, 104].

The risks of voting systems to malicious and non-malicous attacks, rather than operational design, have also been an interest of researchers. The Election Assistance Commission [63] analyzed threats to voting processes in the U.S. for seven voting technology types. Scala et al. [163] identified security threats for mail-in voting processes and offered a relative likelihood score for each to identify the most important threats to address. Fitzsimmons and Lev [72] study geographic-based risks by introducing a control problem and demonstrate that the election results can be manipulated through the strategic selection of polling locations.

Notably absent from the literature are studies into the resiliency of voting systems and analytical techniques that can be used by election administrators to inform design decisions in response to election-related disruptions. We consider a disruption to the voting system to be any event that would cause election administrators to change their normal operating procedures, such as a pandemic, resource shortage, or budget cut. In practice, election administrators have flexibility to modify the voting system in response to a disruption to best serve the voters, even though in many cases the election has been planned well before a disruption occurs. Responses to a disruption can be operational, such as increasing

the distance between voting booths during a pandemic. Others represent infrastructure changes, such as consolidating polling locations or adding ballot drop box locations. There is a need for analytical tools that can help election administrators understand how voting systems can be redesigned cost-effectively in order to minimize the burden place on voters, since even small changes to the voting system can have a substantial impact on voter participation [88, 26]. These tools can help election officials develop contingency plans that can be pre-vetted by the appropriate authorities and allow election officials to be nimble in response to disruptions.

1.3 Dissertation contributions

In this dissertation, we introduce new optimization and simulation models to help election officials respond to voting system disruptions and improve voting system resilience. Using these models, we analyze and evaluate voting system designs and outline implications for practice. While the motivation for these contributions is to improve the resiliency of voting systems, they can also be used to inform the design of cost-effective voting systems during normal operations. Often, these goals are the same. The principal contributions of this dissertation are divided into four chapters and the contributions contained therein are summarized as follows.

In Chapter 2, we study how to design in-person voting systems whose performance are robust to pandemic conditions, such as protective measures implemented during the COVID-19 pandemic using a case study of Milwaukee, WI. We introduce a mathematical model and discrete event simulation of the in-person voting process to capture pandemic-related changes and assess various voting system designs on the voter wait times, voter sojourn times, line lengths, voter time spent inside, and the number of voters inside. The analysis indicates that poll worker shortages, social distancing, personalized protective

equipment usage, and sanitation measures can lead to extremely long voter wait times. We consider several design choices for mitigating the impact of pandemic-related changes on voting metrics. The case study suggests that long wait times can be avoided by staffing additional check-in locations, expanding early voting, and increasing the physical footprint of polling locations. Additionally, the analysis suggests that implementing a priority queue discipline has the potential to reduce waiting times for vulnerable populations at increased susceptibility to health risks associated with in-person voting.

In response to an election-related disruption, election administrators may consider consolidating polling locations. During the COVID-19 pandemic, poll worker shortages and rising election costs led many jurisdictions to take this approach. However, determining when it is appropriate to consolidate polling locations and how to consolidate polling locations if necessary is challenging. In Chapter 3, we formalize the set of constraints and criteria election officials should follow as the polling location consolidation problem (PLCP). We introduce an integer programming model of the PLCP to determine when it is necessary to consolidate polling locations and how to do so optimally. The PLCP simultaneously selects which polling locations to use in the upcoming election, reassigns voter precincts to polling locations, and allocates critical resources to the selected polling locations. The PLCP minimizes the increased distance that voters must travel to their updated polling location. In this way, the PLCP reduces the number of changes made to the voting system and limits the impact on voter participation. Empirical research also demonstrates the importance of reducing the wait time experienced once arriving at a polling location. We require that most voters do not wait longer than a pre-specified limit, such as 30 minutes, using a chance constraint. We show that it is NP-Hard to identify a feasible solution to the PLCP and introduce valid inequalities for the PLCP. Using a case study of Richland County, South Carolina, we demonstrate the value of the PLCP and investigate implications for practice. We find that consolidating a moderate number of

polling locations may be a reasonable approach to mitigating disruptions to the voting system. Moreover, we demonstrate the value of well-located, large buildings that can be used as polling locations to the resiliency of the in-person voting system.

Consolidating polling locations can leave some areas with low accessibility to the in-person voting system. Moreover, increased use of the absentee voting system can substantially increase costs and increase the risks associated with submitting a ballot through the mail. In response, election officials can add new drop boxes to their voting infrastructure to best serve their voters. However, existing techniques for locating drop boxes are limited. In Chapter 4, we introduce an integer programming model, the drop box location problem (DBLP), to optimally locate drop boxes. The DBLP considers criteria of cost, voter access, and risk. The cost of the drop box system is determined by the fixed cost of adding drop boxes and the operational cost of a collection tour by a bipartisan team who regularly collects ballots from selected locations. The DBLP utilizes covering sets to ensure each voter is in close proximity to a drop box and incorporates a novel measure of access to measure a voter's ability to use multiple voting pathways to vote. The DBLP is shown to be NP-Hard, and we introduce a heuristic to generate a large number of feasible solutions for policy makers to select from a posteriori. Using a real-world case study of Milwaukee, WI, we study the benefit of the DBLP. The results demonstrate that the proposed optimization model identifies drop box locations that perform well across multiple criteria and consistently outperform "rules-of-thumb." The results also demonstrate that the trade-off between cost, access, and risk is non-trivial, which supports the use of the proposed optimization-based approach to select drop box locations.

Changes to the voting infrastructure can increase the burden placed on voters [18]. For many individuals, the ease of casting a ballot can be improved through the actions taken by individual spontaneous volunteers or volunteer groups. These include efforts to "get out the vote," inform voters of changes to the voting system, provide a witness signature for

absentee voting, and help potential voters transit to the polls. While the actions taken by spontaneous self-assigning volunteers are beneficial to society, the lack of a central decision maker results in inefficient efforts. We posit that there is an opportunity to generate task recommendations for volunteers in real-time throughout the election period to improve the efficiency of volunteer efforts, while still providing volunteers the freedom to select their desired tasks.

The contents of Chapter 5 were originally published with application to disaster response efforts, however they can be adapted and applied to the outlined elections management problems. In Chapter 5, we study how to indirectly manage spontaneous volunteers who are a critical and often overlooked resource, particularly during disaster response. Frameworks that do consider spontaneous volunteers often rely on the use of a central controller to assign them to tasks, which is not always feasible due to privacy and legal liabilities. We investigate a new paradigm in which volunteers self-assign to tasks from an ordered list of recommendations. We present the problem of generating task recommendations as the Online Task Recommendation (OTR) problem. Subsequently, we present the Maximal Ordered Multiple Matching (MOMM) problem, an integer optimization approach to construct task recommendations for volunteers by matching tasks and volunteers based on the expected volunteer utility from completing each task and the societal value of completing each task. We demonstrate the benefit of the proposed methods with both real-world and synthetic case studies with spatially distributed tasks using Monte Carlo simulations. We find that recommendations generated by the proposed models guide the completion of the most critical tasks by reducing task over-service redundant volunteer responses often associated with spontaneous volunteers.

In addition to contributing new models, theory, and algorithms, the research contained within this dissertation sets the groundwork for continued research and contribution to election practice by the industrial engineering field. However, the practical impact of the

proposed research goes further, since the models serve as the technical basis for election management policies that make election voting systems safer, more reliable, and more equitable.

1.4 Dissertation organization

The remainder of this dissertation is structured as follows. In Chapter 2, we study the resiliency of the in-person voting system to pandemic related changes and assess various mitigating practices. In Chapter 3, we study how to optimally consolidate polling locations. In Chapter 4, we study how to locate ballot drop boxes to minimize cost while meeting requirements related to accessibility and risk. In Chapter 5, we study how to generate recommendations to self-assigning spontaneous volunteers. Finally, I present concluding remarks in Chapter 6.

Chapter 2

Designing pandemic-resilient voting systems

2.1 Introduction

Election officials spend months preparing for an election and implementing strategies to ensure that voting is effective, equitable, accessible, and quick [191]. Often, challenging resource allocation and design decisions must be made prior to Election Day so that voters can cast their ballots without facing long voting queues or disenfranchisement [29]. However, these decisions must be made before the actual conditions on an Election Day are known, which motivates the need to study these systems using statistical models.

The 2020 general election occurred while many states operated under emergency orders related to the COVID-19 pandemic. The COVID-19 pandemic placed unusual strains on the voting system that made design decisions even more challenging. For instance, voting systems are reliant on volunteer poll workers. However, most election volunteers are over the age of 60 and at high-risk to COVID-19 [41, 182]. Election commissions had to consider how to maintain the necessary number of poll workers, while also designing the voting

process such that poll workers (and voters) had minimal exposure to COVID-19 on Election Day [29]. The performance criteria for the 2020 general election expanded to also address the health and safety of voters. This has led to new questions for the design and operation of in-person voting systems including sanitation of shared spaces [217], enforcement of social distancing [217], and mandatory use of personal protective equipment.

The performance of voting systems during the 2020 general election suggests that mitigating the risks associated with COVID-19 without impacting voting systems is extremely challenging. The Wisconsin 2020 spring election and presidential preference primary on April 7, 2020 was the first election in the U.S. with in-person voting to be held during a statewide "stay-at-home" order associated with the pandemic [147]. Many areas in Wisconsin faced a shortage of poll workers [22, 55, 182] causing some cities to consolidate polling locations. The City of Milwaukee held its 2020 spring election at five consolidated polling locations, instead of the standard 182 polling sites, to allow for social distancing and to mitigate the impact of poll worker shortages. The decision to consolidate polling locations drew national attention to the issues surrounding the design of in-person voting processes to mitigate COVID-19 risks. A number of lawsuits followed the Wisconsin 2020 spring election, highlighting the need for more robust planning for the 2020 general election and beyond.

In this chapter, we study how to design and operate in-person voting systems that are robust to pandemic-related disruptions. To support this study, we introduce a discrete event simulation of the in-person voting process and include variations to capture pandemic-related changes. We use a case study of the Milwaukee election system to explore these design decisions. Using this case study, we first quantify to what extend personal protective equipment, sanitation of voting areas, social distancing, and poll worker shortages disrupt voting metrics. Subsequently, we quantify the benefit of various mitigating policies and actions on voting metrics. Specifically, our analysis considers the actions election officials

can take to expand access to early voting, increase poll worker recruitment, add additional ballot scanners, expand the physical footprint of polling locations, implement priority queues for at-risk voters, and consolidate polling locations. The results included within this chapter are intended to support election planning during a pandemic.

Novel aspects of the simulation model and analysis within this chapter include the consideration of health-related metrics, social distancing, personalized protective equipment, and various pandemic-related mitigations (e.g., the consolidation of polling locations). The analysis indicates that the pandemic-related disruptions of poll worker shortages, social distancing, and personalized protective equipment usage and associated protective measures can lead to extremely long voter waiting times, increase the time voters spend inside a polling location, and increase the number of individuals inside a polling location. We find that expanding access to early and absentee voting and increasing poll worker recruitment efforts are critical to mitigating the impact of pandemic-related disruptions. We also find that expanding the physical footprint of polling locations to allow for the distant placement of voting booths can improve voting metrics and that priority queueing for at-risk voters can reduce the health risks associated with voting for the most at-risk populations. We demonstrate that increasing the number of ballot scanners has a minimal impact on voting metrics. Finally, we show that in most cases, consolidating polling locations is not an effective mitigating policy in general, although in some cases it may be necessary.

The remainder of this chapter is organized as follows. Section 2.2 reviews the existing operations management-based elections literature. Section 2.3 introduces a discrete simulation model of the voting process. Section 2.4 describes the case study of Milwaukee, Wisconsin. Section 2.5 discusses the model validation. Sections 2.6 and 2.7 presents the results of the case study. Section 2.8 provides a discussion of case study results and limitations of our analysis.

2.2 Literature Review

The operation of in-person voting systems has drawn the attention of many researchers. Often, multiple criteria are used to evaluate the performance of voting systems [130], and voter wait times in queues has emerged as one of the most important considerations for designing in-person voting systems. It has been argued that the queueing of voters must be taken into account when considering voting access [190]. For example, experts estimate that more than $200,000$ voters in Florida may have been deterred to vote in the 2012 Presidential Election due to long voting queues [155].

Some literature aims to describe the voting process. Stein et al. [186] present a multi-county observational study of voting during the 2016 Presidential election. They investigate the time it takes voters to complete steps within the voting process and how the design of the voting system impacts expected process times. Spencer and Markovits [181] present an observational study of thirty polling location in three California counties and study the attrition rate of voters in a queue, the check-in service rate, the time to vote, and poll worker characteristics.

A stream of papers has focused on the allocation of voting machines to polling locations, since voting machines have been recognized as a bottleneck in the voting process [221]. Methods to allocate voting booths or machines to reduce wait times include simulation [221], queueing theory [6], simulation optimization [222], integer programming [214], a combination of queueing and simulation [223], and robust optimization [224]. Li et al. [112] use simulation optimization to demonstrate that average voting times can be reduced by better allocating voting booths to voting polling locations. Wang et al. [214] examine how to allocate resources such as voting machines that balance trade-offs across equity and efficiency using integer programming models. Yang et al. [224] develop a robust optimization model to study how to mitigate worst-case voting queues. Rather

than allocation, some research focuses on selecting resource types to improve in-person voting systems. Edelstein [64] compares voter waiting times using lever machines, direct recording electronic (DRE) voting machines, and paper ballot scan systems. Edelstein and Edelstein [65] investigate the use of DRE voting machines and paper ballot optical scanners to determine which best mitigates long lines and voter disenfranchisement.

Other research attempts to determine, after the election occurred, if voting system design resulted in voter disenfranchisement. Highton [92] attempts to determine if available voting machine resources caused lower-than-expected turnout in the 2004 general election in Ohio and whether the change in turnout would have changed the election results. Buell et al. [20] audit election results for the 2010 Democratic primary in South Carolina by analyzing audit trail files from DRE machines to determine if some votes were not counted. Buell [21] uses simulation to verify that media reports of a large number of voters waiting in line for several hours and detect the causes of the long waits the 2012 general election in South Carolina. Allen and Bernshteyn [6] use queueing theory to determine that voters were deterred from voting in Franklin County, Ohio in 2004 due to the allocation of voting machines.

Few papers study how to design voting systems in addition to resource allocation decisions. Three broad mechanisms to reduce wait times have been noted in the literature. The first is to reduce the number of voters who come to a polling place [190]. The second is to increase the number of service points, such as check-in booths, voting booths, and ballot scanners [190]. The third is to reduce the average "transaction" time of voters [190]. Stein et al. [186] study how changes in the voting system, such as the use of voting identification requirements, impact voting times. Bernardo et al. [16] use a discrete event simulation to investigate the impact of separating provisional voters at check-in and find that doing so improves the average time voters spend in the system.

The 2020 COVID-19 pandemic motivates the need to consider the impact of methods

to mitigate disease transmission and protect patient safety during elections [147]. To our knowledge, there are no studies that have investigated the operation of voting systems in terms of the health and safety of voters.

2.3 Simulation modeling approach

We introduce a discrete event simulation model of the in-person voting process. In most elections, voters are assigned to one polling location to which they travel and cast a vote on an Election Day. This is achieved by partitioning an area into geographic regions called voting precincts or wards. We describe and model the operation of the voting process at a polling location. The performance of individual polling locations can then be aggregated to give a representation of the performance within a region. While each election jurisdiction may have a slightly different in-person voting process, we introduce a model that represents a typical in-person voting process [186] and later describe modifications to it.

Figure 2.1 illustrates the different steps and components of the in-person voting process. First, (A) the voter enters the voting system and enters a queue to check-in. Then, once poll worker(s) are available, (B) the voter registers (if necessary), checks-in, and obtains their ballot. Then, (C) the voter enters a queue for a voting booth, and once a voting booth is available, (D) the voter marks their ballot. The voter then (E) enters a queue to submit their ballot to an optical ballot scanner. Lastly, once a ballot scanner becomes available, (F) the voter leaves the queue and submits the ballot in the optical ballot scanner. Once the ballot is approved, the voter leaves. In addition to the described steps, (G) a poll worker may need to clean or sanitize the voting booth after a voter marks their ballot, and a voter cannot use that booth until sanitation is complete. This step is typically not done during a normal election. The queues (in steps A, C, and E) are traditionally first-come, first-served (FCFS) queues. Within a polling location, there are three main resources: check-in booths (step

B), voting booths (step D), and optical ballot scanners (step F). The number of resources shown in Figure 2.1 are for illustration purposes and may vary at each polling location. The logic to create a computer simulation of this process is provided in Figure 2.2.

We assume each polling location has a *maximum capacity* defining the number of voters that can be inside, after poll workers are accounted for. We define the number of voters inside as the total number of voters in steps B, C, D, E, and F. If the number of voters within the polling location is equal to the maximum capacity, then no new voters can progress from queue A to process B until a voter already inside completes the voting process and leaves the polling location.

In a given jurisdiction, the in-person voting process of all polling locations are likely to follow the same logic, but the inputs (e.g., number of voting booths in D) may vary. Some polling locations may use electronic voter poll books within step B while other use paper books. Some polling locations may use electronic voting machines, such as direct-recording electronic voting machine, in step D rather than votes being cast on a paper ballot. In this case, steps D and F are merged, paper ballots and voting booths are no longer used, and the queue E no longer exists (except to get "I Voted" stickers). Many jurisdictions also use ballot marking devices. Different electronic voting machines may have different processing time and user-error rates that require help from a poll worker. The optical ballot scanner in step F may also vary by location, when a ballot scanner is used. Different ballot scanners have different functionalities, different processing times, and different reliability of functioning on an Election Day. Some locations may not use an optical ballot scanner and instead count votes by hand. The same model of the system can be used with appropriate inputs for different steps of the process to represent the setting at hand.

There are many metrics used to evaluate the in-person voting system during a non-pandemic election, and there are trade-offs between many of these metrics [130]. Some well-known metrics include voter wait time, voter sojourn time, voter turnout, polling

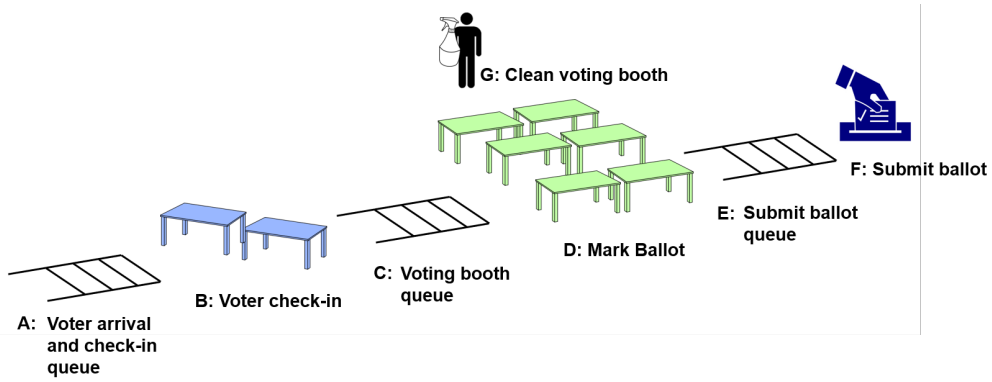


Figure 2.1: A model of the in-person voting process

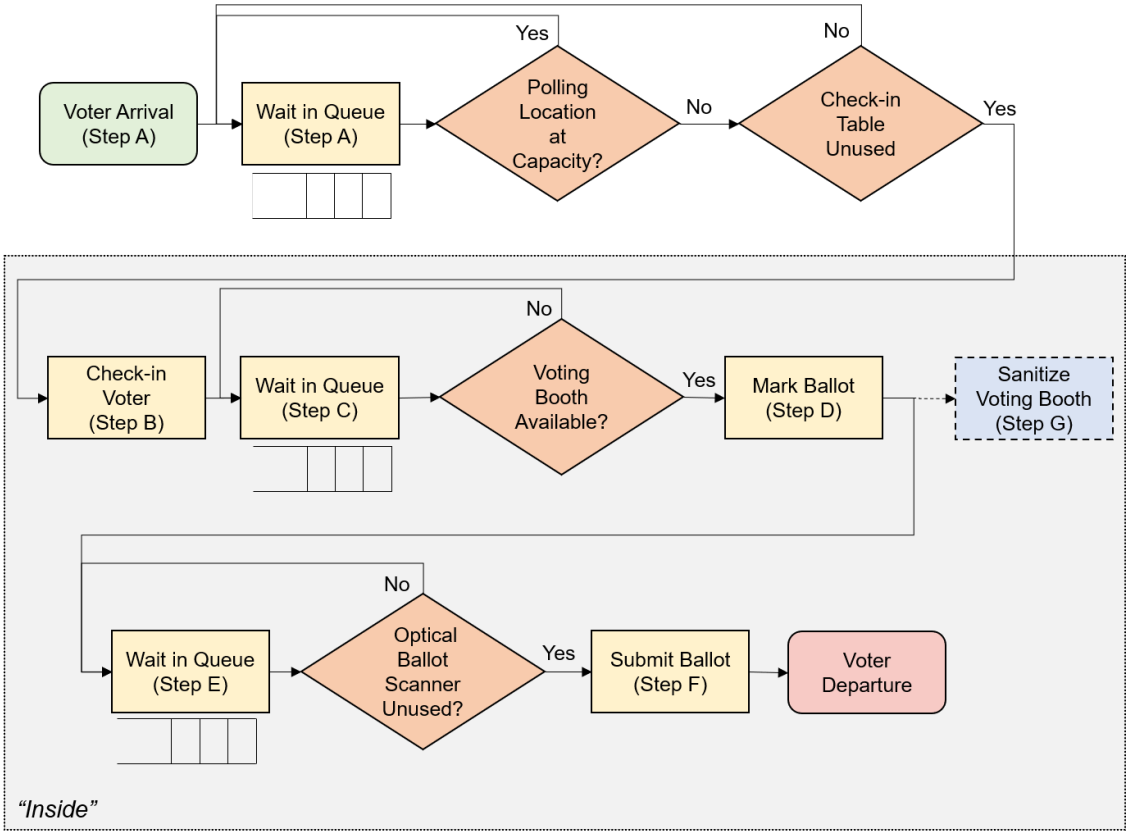


Figure 2.2: Logic used in the computer simulation of a polling location.

location line length, and ballot rejection rate [130]. Voter wait times is a major area of focus. The Presidential Commission on Election Administration has set a goal that no citizen should have to wait more than 30 minutes to vote [149]. Some areas have more stringent

goals, such as no voter waiting more than 15 minutes [14]. It is also expected that polling locations are close to the voters who are expected to use them so that the "cost" to vote is low. In addition to these metrics, the COVID-19 pandemic has caused election officials to consider health-related metrics, including the amount of time spent inside (especially for high-risk voters) and the number of voters inside, since they are indicators for the risk of COVID-19 transmission [29]. These metrics apply to many other health-related risks. While not explicitly studied within this research, these metrics can be used to estimate disease transmission rates, but rates depend on the setting and disease. Table 2.1 describes the performance metrics of interest for all in-person (Election Day) voters and polling locations that are considered in this study.

Table 2.1: In-person voter and polling location metrics and their descriptions.

In-person Voter Metrics

Metric	Unit	Description
Average Wait Time	Minutes	Average time spent in A
Average Time Inside	Minutes	Average time spent in steps B, C, D, E, and F
Average Sojourn Time	Minutes	Average time spent in steps A, B, C, D, E, and F
15 Minute Wait	N/A	Proportion of voters spending 15 minutes or more in A
30 Minute Wait	N/A	Proportion of voters spending 30 minutes or more in A
Distance to Poll	Miles	The distance between the voter and polling location

Polling Location Metrics

Metric	Unit	Description
Average Line Length	Voters	Average number of voters in A
Average Voters Inside	Voters	Average number of voters in steps B, C, D, E, or F

2.3.1 Pandemic-related disruptions

While the introduced model of the in-person voting system captures the normal operations of the in-person voting process at each polling location, a pandemic changes how the

in-person voting system can operate. We briefly describe how the pandemic-related disruptions of personal protective equipment and sanitation requirements, social distancing, and poll worker shortages impact the in-person voting process at a polling location. We investigate the impact of these changes in the subsequent sections.

Personal Protective Equipment and Sanitation Requirements (PPE)

In previous elections, apart from those held during the COVID-19 pandemic, a minimal amount of personal protective equipment has been used and minimal sanitation of voting space has been conducted. For the 2020 general election, the Centers for Disease Control and Prevention (CDC) suggested rigorous personal protective equipment usage and sanitation policies at all polling locations to mitigate health risks [29]. This may also be implemented in response to future public health emergencies. We assume that personal protective equipment usage and sanitation affect the voting process in two ways. First, the average time to check-in a voter in step B increases due to protective methods, such as to ensure the voter has appropriate personal protective equipment (e.g., a mask), verify the voter's photo-identification while the voter wears a face covering, or ensure adequate spacing during the check-in process. Second, step G of the voting process introduced in Figure 2.1 is activated and a poll worker must sanitize a voting booth once a voter finishes using it.

Social Distancing (SD)

As was seen during the COVID-19 pandemic, social distancing can be implemented to reduce the likelihood of airborne disease spread and protect the health of voters and poll workers. We consider the impact of social distancing in two ways. First, we assume that the maximum capacity of the polling location is reduced. Second, to ensure proper space between voters, the number of voting booths is reduced. We capture the extent to which the maximum capacity and number of voting booths is reduced through the use of a *capacity*

factor. The capacity factor defines the proportion of the normal level of the maximum capacity and number of voting booths (rounded up) that are retained within the polling location. For example, if there are typically 9 voting booths and a maximum capacity of 31 voters in a polling location, a capacity factor of 50% allows for 5 voting booths and a maximum capacity of 16 voters.

Poll-worker Shortages (PWS)

Poll worker shortages can result from additional responsibilities to manage the in-person voting process (e.g., managing unusually long lines, sanitizing voting areas) with the same number of poll workers or a lower-than-usual number of poll workers. We capture the impact of poll worker shortages by reducing the number of check-in booths that can be staffed. However, at the same time, a reduction in the number of check-in booths increases the maximum capacity of voters by the number of poll workers typically staffing a check-in booth.

2.4 Case study: Milwaukee, Wisconsin data

We create a case study for the City of Milwaukee, Wisconsin. We first describe the details of the voting process in Milwaukee during a non-pandemic election and then describe pandemic-specific details.

Milwaukee is the largest city in the state of Wisconsin with an estimated 590,157 residents [202]. Due to the COVID-19 outbreak, election officials placed an increased emphasis on ensuring the health and safety of voters, particularly in-person voters who are vulnerable to risks associated with in-person voting. Age is the single largest risk factor for severe illness from COVID-19 [30] and for many other diseases. An estimated 13.8% of the voting age population in Milwaukee is 65 or older [202]. Later, we use this as an estimate for

the fraction of in-person voters who are "high-risk" to pandemic-related health issues. According to the CDC, individuals younger than 65 years of age with preexisting conditions may be at high-risk to COVID-19 [30], which would increase the percentage of voters who are high-risk. However, we assume that a disproportionate number of high-risk individuals use early/absentee voting methods or do not self-identify as high-risk for in-person voting, leading to 13.8% of in-person voters on the 2020 Election Day self-identifying as high-risk.

The City of Milwaukee has 327 voting wards, which have an official assignment to one of the 182 standard polling locations throughout the city [37, 127]. This assignment is made by the Milwaukee Election Commission (MEC). To estimate the voting age population in each of the voting wards, we use the 2011 voting age population in each ward, which is the most recent report provided by the city [37]. Between 2010 and 2019, there was an estimated population change of -0.72% in Milwaukee [66], so we assume the 2011 population in each ward is a reasonable estimate for the voting age population in 2020.

The details surrounding the specifics of the in-voting process at each polling location within our case study design are as follows. The distributions of the process times for each step of the in-person voting process at each polling location is summarized in Figure 2.3. These distributions are fitted to the best available data to represent voting within Milwaukee.

Step A. Voters arrive according to a non-stationary Poisson process and they enter a queue to check-in to vote, which follows a first-come, first-served (FCFS) policy. The interarrival times are therefore exponentially distributed with a rate of $r = \frac{f \times p \times v \times (1-e)}{30}$ where f is the fraction of voters arriving during the 30 minutes period, p is the voting age population assigned to the polling location, v is the expected voter turnout, and e is the proportion of voters who vote early or absentee. The percentage of voters, on average, that arrive during each 30 minute interval is shown in Figure 2.4. This distribution has been used in previously published

research [223], and is the only research known to the authors to report the numerical arrival rate at such granularity. In reality, arrival rates at polling locations may be different due to socioeconomic status; however, we assume a similar pattern is observed at each polling location. We assume that the early voting rate and expected voter turnout is the same in all voting wards; however, the actual turnout and arrival pattern of voters at each polling location vary within the simulation due to stochasticity within the arrival process. We also assume that the number of individuals who vote in every ward is independent of the polling location to which the ward is assigned.

Step B. Each polling location has a receiver and registration table for each ward assigned to the polling location, plus additional tables where voter registration is higher (Personal Communication with MEC 4/29/2021). To account for this, we add 36 tables (approximately 20% of 182) to the polling locations with the highest ratio of voting age population to the number of tables. For the remainder of this chapter, we refer to the receiver and registration tables as "check-in booths." Each check-in booth is staffed by two poll workers who use paper registration lists. The time to check in a voter follows the lognormal distribution in Figure 2.3a with an average of 1.85 minutes. This distribution is based on research conducted by Stein et al. [186] of polling locations across the country, including a polling location in Dane County, WI. They stratify results by key structural designs (e.g., same day registration) that we use to identify an appropriate distribution that also leads to simulation outputs that match known values for election metrics. In reality, there are two classes of voters: those that are already registered and those who have not yet registered. Voters in the two classes are likely to have different expected lengths of time at a check-in booth. However, to the knowledge of the

authors, there is no publicly available data or study that provides estimates for the check-in time stratified by voter type.

Step C. The queue for a voting booth is a FCFS queue.

Step D. We estimate the number of voting booths at each polling location using the voting age population assigned to vote at a polling location. We assume each polling location has $\lceil 2 \times 4.038 \times voting\_age\_population/(13 \times 60) \rceil$ voting booths, where 2 represents a scaling factor to address fluctuations in the arrival rate of voters, 4.038 represents the average number of minutes to vote [223] and 13×60 represents the number of minutes during the voting period. The actual number of voting booths at each polling location in Milwaukee was not available from the MEC (Personal Communication with MEC 4/29/2021). The distribution of voting time follows the lognormal distribution described in Figure 2.3c, which is adopted from previously published elections research [223].

Step E. The queue to submit a ballot to an optical ballot scanner is FCFS.

Step F. Most polling locations have one DS200 optical ballot scanner and a small number of sites have two (Personal Communication with MEC 4/29/2021). In our study, we assume each location has a single scanner. Our results indicate that voting metrics are unlikely to substantially change with an additional ballot scanner, since the utilization of the ballot scanners is relatively low. The time to submit a ballot is captured by the distribution in Figure 2.3e. This distribution is determined using publicly available training videos of DS200 machines.

We assume that the size of each polling location is selected as to appropriately fit the needed number of resources, voting booths, and queues. As a result, we set the "typical" maximum capacity at each polling location to be equal to the sum of the number of check-in

booths, two times the number of voting booths (to account for queues), and the number of optical ballot scanners. This equation was calibrated to information provided by the MEC (Personal Communication with MEC 4/29/2021).

2.4.1 Pandemic-related disruption inputs

We briefly introduce the inputs to the simulation model that are specific to pandemic-disrupted elections.

Personal Protective Equipment and Sanitation Requirements (PPE)

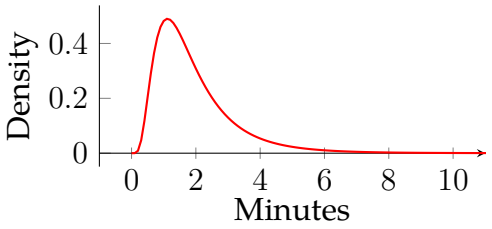
Personal protective equipment and sanitation requirements impact the in-person voting process in two ways. First, the time to check-in a voter increases. We assume the average time to check-in increases by 20 seconds. The distribution of time to check-in a voter when personal protective equipment and sanitation are required is illustrated in Figure 2.3b. These requirements also lead to a sanitation step, step G, in the voting process. We assume the time to sanitize a voting booth from the moment a voter leaves the voting area to when a new voter can use the voting booth follows a distribution as illustrated in Figure 2.3d. A poll worker must recognize the voting area needs to be cleaned, walk to the location, sanitize the location, and then indicate that the booth has been sanitized.

Social Distancing (SD)

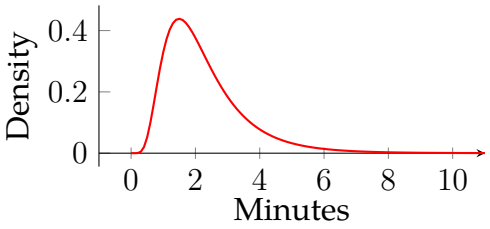
Social distancing reduces both the number of voting booths at a polling location and the maximum capacity of the polling location. During an election where social distancing is enforced, we set the capacity factor to 25% (from 100%). This is based on the CDC recommended social distancing distance of a 6 foot radius around each voter, which requires 4 times the space as a 3 foot radius, which we use to represent the typical "personal space" of a voter.

Poll-worker Shortages (PWS)

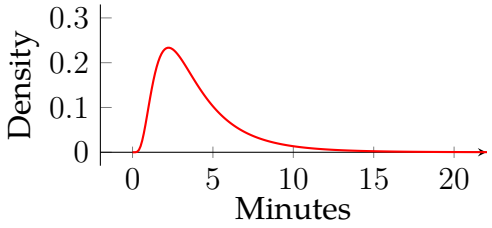
A poll worker shortage can be caused by either a lower than normal poll worker turnout or the same poll worker turnout with additional tasks to complete, as might be expected during a pandemic. In both cases, we assume that polling locations remove one check-in booth in response to poll worker shortages, since the booths cannot be staffed.



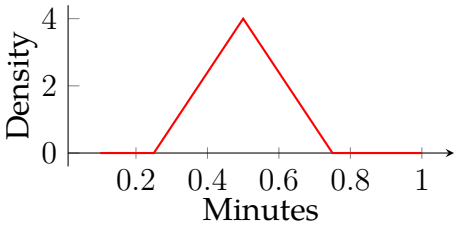
(a) Check-in Duration (No PPE)
lognormal(*0.478*,*0.607*)



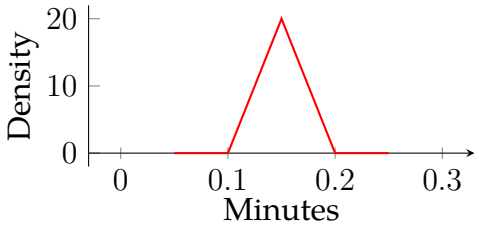
(b) Check-in Duration (PPE)
lognormal(*0.681*,*0.530*)



(c) Mark Ballot Duration
lognormal(*1.199*,*0.627*)



(d) Clean Voting Area Duration
triangular(*.25*,*0.5*,*0.75*)



(e) Submit Ballot Duration
triangular(*0.1*,*0.15*,*0.2*)

Figure 2.3: Probability distributions describing the time to complete each step of the voting process.

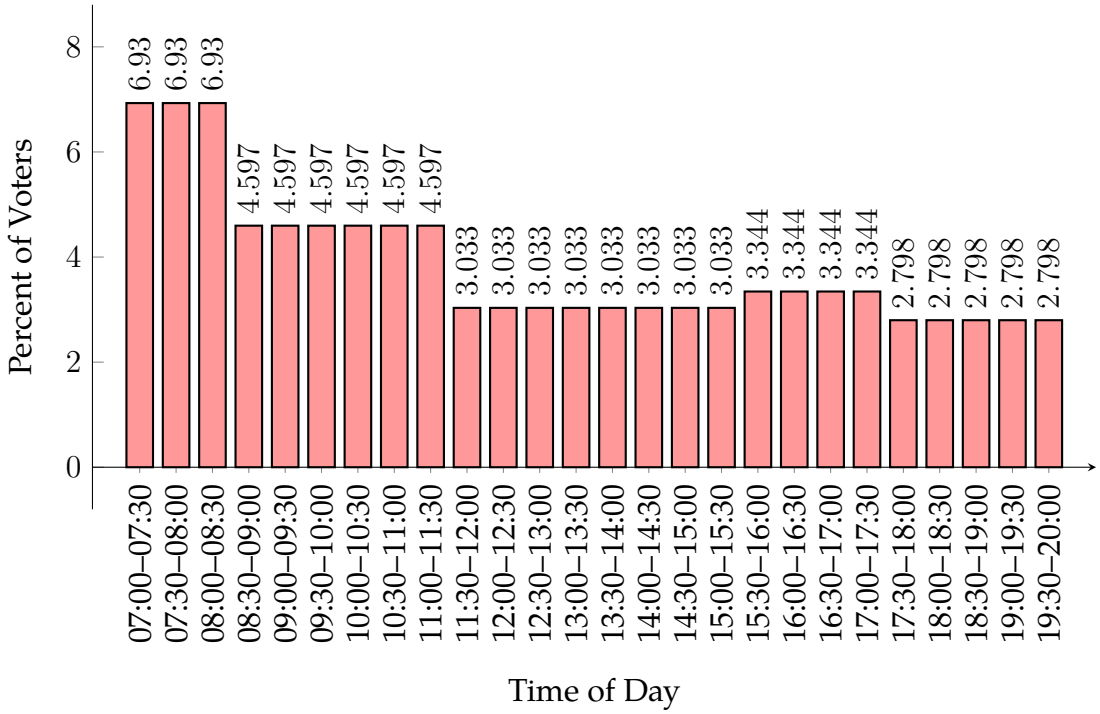


Figure 2.4: Percent of voters arriving during each 30 minute increment on average.

2.5 Model Validation

We took a three step approach to model validation. First, we presented the simulation model to a professor of Political Science at the University of Wisconsin-Madison with expertise in Wisconsin elections to establish face validity. Second, we discussed the structural and data assumptions used for the case study simulation with this same expert. Third, we ran the simulation using inputs from three previous elections to compare the simulation model outputs to the known outputs from the actual in-person voting system, given the conditions in which the elections were held. These elections include the 2016 general election, the 2020 spring election, and the 2020 general election. It is not common practice to record voting metrics during an Election Day at each polling location, unless reported by the media. As a result, we use publicly available media reports and research studies where possible to validate our model. We acknowledge this is an area for future work

when such a process can be done safely. Considering the very nature of health risks of observing elections during a pandemic and the importance of releasing such information to the public in time to be useful, we validated the model to the extent existing data allow. The results presented in this and the remaining sections were obtained from a simulation implemented in Python.

2.5.1 2016 General Election

In November 2016, the City of Milwaukee [128] had a 57.2% voter turnout given the number of voting age individuals we consider. In Milwaukee County, 29.6% of ballots cast were using early/absentee methods [216, 215]. This was the most granular data reported, so we use this rate for the City of Milwaukee. We run the discrete event simulation for each of the 182 polling locations and aggregate the results. The mean values and standard deviations of seven voting metrics across fifty replications of the simulation are presented in Table 2.2.

To validate our model, we compare the outputs from our simulation to an analysis that uses smart phone data to estimate the length of time voters spent at their polling locations during the 2016 Election Day [33]. Most metrics were reported at a national level, but some were reported for Wisconsin's 4th district, which includes the City of Milwaukee [33]. The 95% confidence interval for the mean average sojourn time output by the simulation is 15.6 ± 0.9 minutes, which includes the estimated sojourn time of 15.50 minutes (Bayesian adjusted estimate of 15.58) by Chen et al. [33] for Wisconsin's 4th district. Chen et al. [33] also estimate the maximum sojourn time experienced by a voter in the United States at 119.50 minutes. This was the most granular data reported. The 95% confidence interval for the mean maximum sojourn time output by our simulation is 115.0 ± 4.4 minutes which falls below the estimated maximum wait time by Chen et al. [33] for the United States. However, their estimate for the entire United States has an average sojourn time of 19.3 minutes, which is larger than the estimated average sojourn time experienced in Wisconsin's 4th

district (15.5 minutes). Therefore, it is also reasonable to expect the maximum wait time to be slightly higher. Chen et al. [33] also estimate that 18% of voters in the United States had a sojourn time of 30 minutes or more. This was the most granular data reported. In our simulation $15.6\% \pm 1.5\%$ of voters experience a sojourn time of 30 minutes or more. This is lower than the reported data [33], and this discrepancy is likely due to a lower average sojourn time in Wisconsin's 4th district than the entire United States.

Table 2.2: Mean value (standard deviation) of in-person voting metrics for the 2016 general election across 50 replications.

Avg. Wait Time	Avg. Time Inside	Avg. Sojourn Time	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
9.2 (3.1)	6.5 (0.1)	15.6 (3.1)	0.21 (0.06)	0.12 (0.04)	11.3 (4.2)	7.9 (0.4)

2.5.2 2020 Spring Election

The 2020 spring election held in April 2020 was the first election held in the state of Wisconsin since the onset of the COVID-19 pandemic. The City of Milwaukee [129] had a 22.0% voter turnout given the number of voting age individuals we consider. In Milwaukee County, 80.5% of the ballots were cast early [220]. This was the most granular data reported, so we use this rate for the City of Milwaukee.

During the 2020 spring election, social distancing (SD) was implemented and personal protective equipment and sanitation were required (PPE). In addition, the Milwaukee Election Commission faced poll worker shortages and decided to consolidate polling locations in response to this disruption. The City of Milwaukee used five polling locations to which residents in the 327 wards were assigned [28]. According to available accounts, the number of optical ballot scanners and check-in booths were limited due to space availability and poll worker shortages [28, 47]. To capture this, we set the number of check-in booths

at each polling location to ten and the number of ballot scanners to five. The mean values and standard deviations of seven voting metrics across fifty replications are presented in Table 2.3.

There was no effort to collect voter metrics during the election in Milwaukee (Personal Communication with MEC 4/29/2021) or to estimate these using other techniques. Thus, we validate our model to publicly available personal accounts reported by news media of voter wait time. According to media reports, there were long lines and voters waited on average 1.5 – 2.5 hours to vote [47, 24, 183, 193]. The simulation estimates an average wait time of 110.4 ± 15.2 minutes to voter which align with the media reports. Our simulation also estimates an average line length of 471.5 ± 66.3 voters at each of the five polling locations, which aligns with reports of long queues.

Table 2.3: Mean value (standard deviation) of in-person voting metrics for the 2020 spring primary across 50 replications.

Avg. Wait Time	Avg. Time Inside	Avg. Sojourn Time	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
110.4 (53.4)	6.6 (0.0)	117.0 (53.4)	0.89 (0.16)	0.84 (0.18)	471.5 (233.4)	30.0 (1.3)

2.5.3 2020 General Election

The 2020 general election held in November 2020 was the first presidential election held in the state of Wisconsin since the onset of the COVID-19 pandemic. The City of Milwaukee [126] had a 57.1% voter turnout given the number of voting age individuals we consider. In the City of Milwaukee, 68.4% ballots were cast early [219]. According to the MEC, COVID-19 safety procedures were implemented to limit the number of people who were able to enter some polling locations to ensure social distancing. We also assume personal protective equipment and sanitation requirements were put in place. Reports indicated

that there was not a poll worker shortage compared to the typical number of workers [171]. We assume the standard 182 polling locations were used. The mean values and standard deviations of seven voter metrics across fifty replications are presented in Table 2.4.

There was no effort to collect voter metrics during the election in Milwaukee (Personal Communication with MEC 4/29/2021) or to estimate these using other techniques. In addition, there were no media reports of long wait times in the City of Milwaukee, suggesting that wait times were relatively short. Our simulation estimates the average wait time experienced during the 2020 general election to be 0.2 ± 0.04 minutes. According to the MEC, no polling location reached the capacity limit in November of 2020 (Personal Communication with MEC 4/29/2021). Our simulation estimates that polling locations were at capacity $0.56\% \pm 0.16\%$ of the time, or just under 4.4 minutes. In 8 of the 50 replications, the polling locations were at capacity an average of 0.1% (47 seconds) of the time or less. This suggests that a capacity factor of 25% may be more restrictive than what was implemented during the 2020 general election. Less restrictive capacity factors do not lead to substantial changes in any of the voting metrics.

Table 2.4: Mean value (standard deviation) of in-person voting metrics for the 2020 general election across 50 replications.

Avg. Wait Time	Avg. Time Inside	Avg. Sojourn Time	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
0.2 (0.1)	6.7 (0.1)	6.9 (0.2)	0.00 (0.00)	0.00 (0.00)	0.1 (0.1)	4.0 (0.2)

2.6 Case Study Results: Impact of pandemic-related disruptions

At the time of writing, the 2016 general election is the most recent presidential election not held during a pandemic. Thus, the simulation outputs for this election serve as a baseline to which we can analyze the impact of social distancing (SD), personal protective equipment and sanitation (PPE), and poll worker shortages (PWS) for future elections. The simulation is run using the voter turnout (57.2%) and early voting rate (29.6%) experienced in the 2016 general election, and apply these disruptions to the voting process. Table 2.5 summarizes the two levels of PPE, SD, and PWS considered in this analysis. For each scenario, we ran 50 replications of the simulation.

Table 2.5: Variables and values used within the full factorial analysis.

		Election Process Inputs	
Disruption	ID	ID = 0	ID = 1
PPE and Sanitation Policies	PPE	Not required	Required
Social Distancing	SD	Capacity Factor = 100%	Capacity Factor = 25%
Poll worker shortage	PWS	Not experienced	Experienced

Figures 2.5, 2.6, and 2.7 present the histograms of the metrics across the 50 replications when each disruption is experienced independently. When the disruption impacts the election, we let the variable equal one (e.g., PPE = 1). When the disruption does not occur, the variable equals zero (e.g., PPE = 0). Figure 2.5 presents the histograms of the value of six voting metrics in the 50 replications of the simulation during an election with no pandemic-related disruptions (- - -) and when PPE=1 (——). The histogram of each metric shifts to the left indicating an increase in the metric. The histograms for the average wait time and average line length also widen indicating that the value of the voting metrics during a single replication is less predictable given the same inputs.

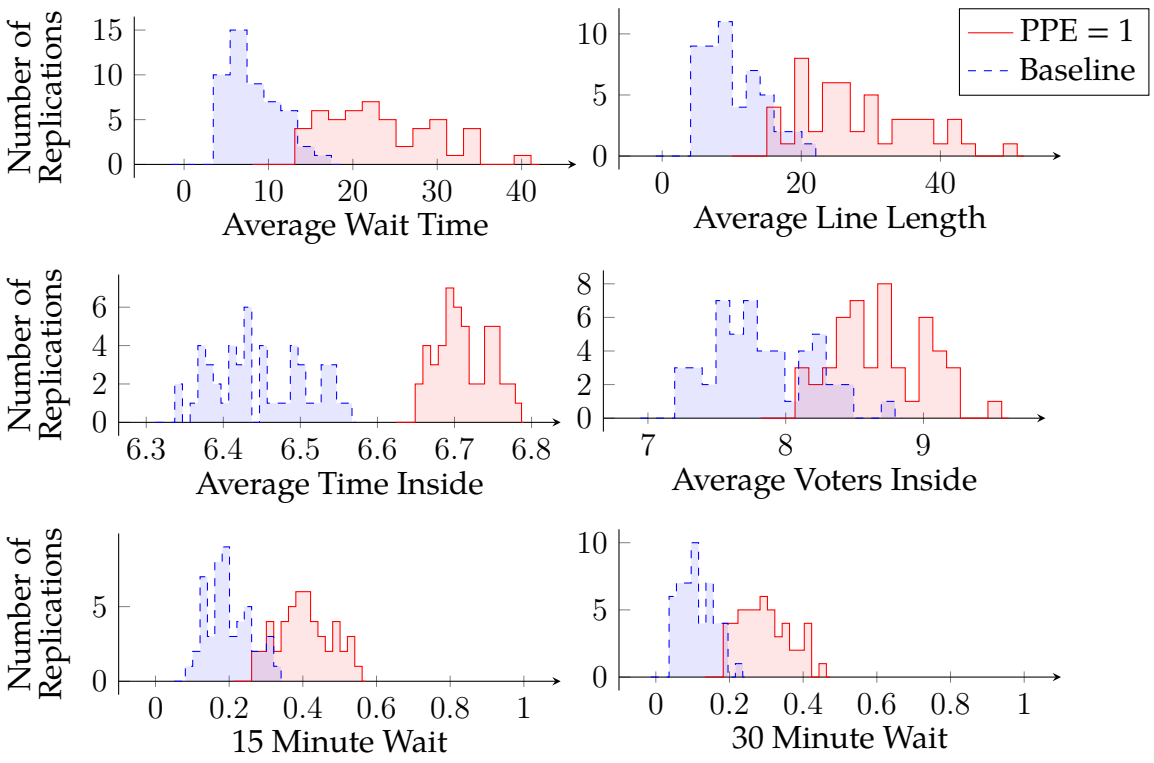


Figure 2.5: Histogram of voting metric values across 50 replications for a baseline election and when PPE and sanitation policies implemented.

Figure 2.6 presents the histograms of the value of six voting metrics across the 50 replications of the simulation during an election with no pandemic-related disruptions (- - -) and when PWS=1 (——). The histogram of each metric shifts to the left indicating an increase in the metric. The histograms for the average wait time, average line length, average time inside, and average voters inside also widen indicating that the value of the voting metrics during a single replication is less predictable given the same inputs.

Figure 2.7 presents the histograms of the value of six voting metrics across the 50 replications of the simulation during an election with no pandemic-related disruptions (- - -) and when SD=1 (——). The histogram of each metric shifts to the left indicating an increase in the metric. The histograms for all metrics also widen indicating that the value of the voting metrics during a single replication is less predictable given the same inputs.

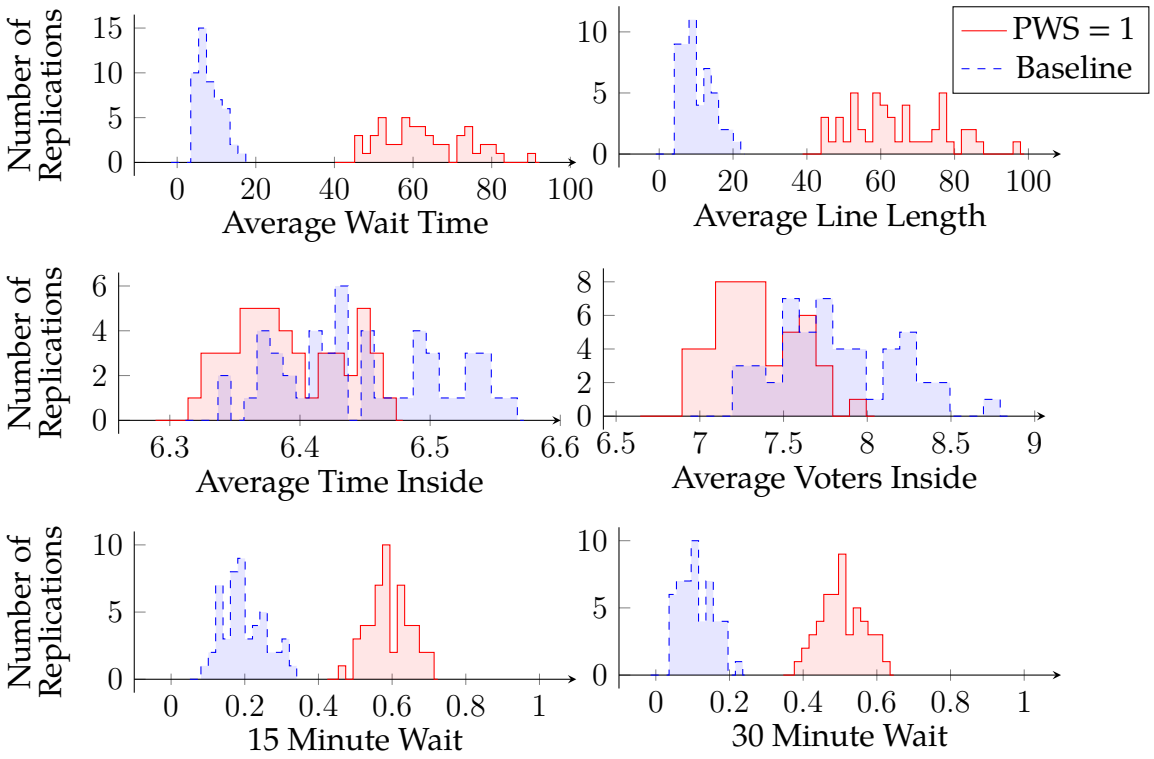


Figure 2.6: Histogram of voting metric values across 50 replications for a baseline election and when poll worker shortage (PWS) is experienced.

We conduct a full factorial analysis for PPE, SD, and PWS. Table 2.6 presents the coefficient values for each factor included within the analysis obtained from a linear regression for the value of six voting metrics as output by the simulation. Also presented are the p-values of the coefficient, which indicate the significance of the coefficient, where p-values less than 0.05 are considered significant. The row for the variable 'Constant' indicates the estimated metric values during the 2016 general election with no pandemic-related disruptions. The values of the coefficients on the first order terms (PWS, PPE, SD) indicate the absolute change in each metric when the disruption is experienced independently. For example, a poll worker shortage (PWS) is associated with an increase of 54.5 minutes in the average wait time. The values of higher order terms (e.g., PWS*PPE, PWS*PPE*SD) capture the impact of multiple disruptions in an election simultaneously. When these terms

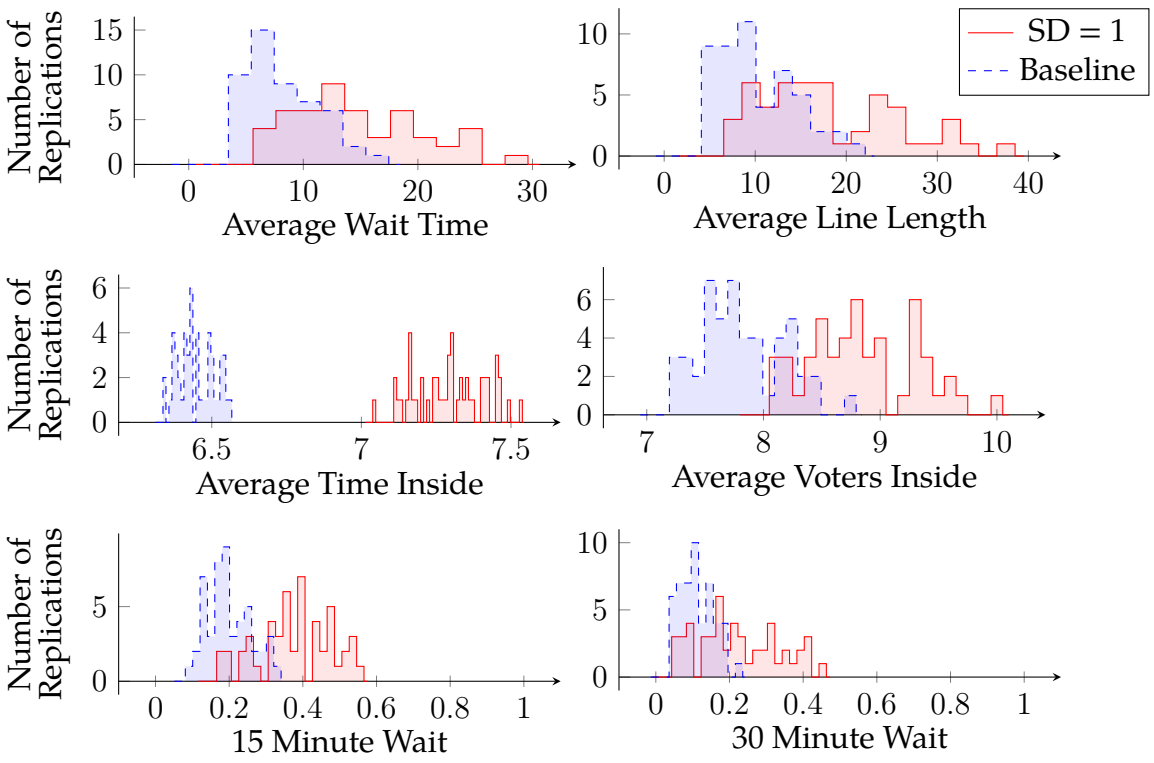


Figure 2.7: Histogram of voting metric values across 50 replications for a baseline election and when social distancing (SD) is enforced.

are positive, there is a compounding effect of the disruptions on the election process. For example, when PWS and PPE disruptions are experienced (PWS*PPE = 1), the expected average wait time increases by $54.5 + 15.1 + 32.7$ minutes. When these terms are negative, such as PWS*SD, the opposite effect occurs. For example, when PWS and SD disruptions are experienced (PWS*SD = 1), the expected average wait time increases by $54.5 + 6.5 - 4.1$ minutes. We find that almost all coefficient terms are statistically significant at a level of 0.05 for each of the voting metrics.

Tables 2.7 and 2.8 present the regression coefficients and p-values for pandemic-related disruptions with voter turnout rates of 47.2% and 67.2%, respectively. These are interpreted the same as Table 2.6. When voter turnout is 47.2%, the impact is of pandemic-related disruptions is less than when the voter turnout rate is 57.2%. However, the relative mag-

Table 2.6: Ordinary least squares regression coefficients (p-values) for each voting metrics when various disruptions are experienced (ID=1) or not (ID=0). The expected voter turnout used is 57.2% and the early voting rate 29.6%.

Variable	Coefficient (p-value*)					
	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant	9.2 (0.00)	6.5 (0.00)	0.21 (0.00)	0.12 (0.00)	11.3 (0.00)	7.9 (0.00)
PWS	54.5 (0.00)	-0.1 (0.00)	0.39 (0.00)	0.40 (0.00)	53.9 (0.00)	-0.5 (0.00)
PPE	15.1 (0.00)	0.3 (0.00)	0.21 (0.00)	0.19 (0.00)	17.9 (0.00)	0.9 (0.00)
SD	6.5 (0.00)	0.8 (0.00)	0.18 (0.00)	0.11 (0.00)	8.0 (0.00)	1.0 (0.00)
PWS*PPE	32.7 (0.00)	-0.0 (0.20)	-0.05 (0.00)	-0.01 (0.44)	20.8 (0.00)	-0.6 (0.00)
PWS*SD	-4.1 (0.09)	-0.2 (0.00)	-0.12 (0.00)	-0.07 (0.00)	-5.1 (0.08)	-0.3 (0.00)
PPE*SD	4.5 (0.02)	0.4 (0.00)	0.02 (0.43)	0.08 (0.00)	5.5 (0.03)	0.5 (0.00)
PWS*PPE*SD	-3.3 (0.44)	-0.2 (0.00)	-0.02 (0.53)	-0.06 (0.05)	-4.0 (0.42)	-0.2 (0.08)

*Using heteroscedasticity robust (HC0) standard errors

nitude of primary and interaction effects are the similar to when the voter turnout rate is 57.2%. Conversely, when the voter turnout rate is 67.2%, the impact of pandemic-related disruptions is larger than when the voter turnout rate is 57.2%.

Table 2.7: Ordinary least squares regression coefficients (p-values) for each voting metric when various disruptions are experienced (ID=1) or not (ID=0). Expected voter turnout used is 47.2% and early voting rate 29.6%.

Variable	Coefficient (p-value*)					
	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant	2.6 (0.00)	6.3 (0.00)	0.06 (0.00)	0.01 (0.00)	2.6 (0.00)	6.4 (0.00)
PWS	25.6 (0.00)	-0.0 (0.00)	0.32 (0.00)	0.27 (0.00)	23.2 (0.00)	-0.1 (0.02)
PPE	5.0 (0.00)	0.3 (0.00)	0.12 (0.00)	0.08 (0.00)	5.1 (0.00)	0.8 (0.00)
SD	1.5 (0.00)	0.6 (0.00)	0.05 (0.00)	0.01 (0.13)	1.6 (0.00)	0.6 (0.00)
PWS*PPE	22.9 (0.00)	-0.0 (0.96)	0.07 (0.00)	0.11 (0.00)	17.3 (0.00)	-0.2 (0.01)
PWS*SD	-1.0 (0.50)	-0.2 (0.00)	-0.03 (0.12)	-0.00 (0.83)	-1.0 (0.50)	-0.2 (0.09)
PPE*SD	1.7 (0.05)	0.3 (0.00)	0.06 (0.01)	0.03 (0.02)	1.7 (0.07)	0.3 (0.01)
PWS*PPE*SD	-1.2 (0.66)	-0.2 (0.00)	-0.04 (0.16)	-0.02 (0.33)	-1.2 (0.66)	-0.2 (0.25)

*Using heteroscedasticity robust (HC0) standard errors

Table 2.8: Ordinary least squares regression coefficients (p-values) for each voting metric when various disruptions are experienced (ID=1) or not (ID=0). Expected voter turnout used is 67.2% and early voting rate 29.6%.

Variable	Coefficient (p-value*)					
	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant	24.5 (0.00)	6.6 (0.00)	0.42 (0.00)	0.31 (0.00)	34.5 (0.00)	9.4 (0.00)
PWS	87.6 (0.00)	-0.1 (0.00)	0.34 (0.00)	0.38 (0.00)	87.9 (0.00)	-1.2 (0.00)
PPE	30.0 (0.00)	0.2 (0.00)	0.20 (0.00)	0.22 (0.00)	36.1 (0.00)	0.6 (0.00)
SD	17.0 (0.00)	1.1 (0.00)	0.26 (0.00)	0.26 (0.00)	24.4 (0.00)	1.5 (0.00)
PWS*PPE	34.4 (0.00)	-0.0 (0.03)	-0.09 (0.00)	-0.10 (0.00)	13.9 (0.00)	-0.7 (0.00)
PWS*SD	-10.9 (0.00)	-0.3 (0.00)	-0.17 (0.00)	-0.17 (0.00)	-15.8 (0.00)	-0.5 (0.00)
PPE*SD	10.3 (0.00)	0.5 (0.00)	0.00 (0.96)	0.02 (0.31)	12.2 (0.01)	0.4 (0.00)
PWS*PPE*SD	-7.7 (0.17)	-0.2 (0.00)	-0.01 (0.54)	-0.02 (0.32)	-9.5 (0.19)	-0.2 (0.14)

*Using heteroscedasticity robust (HC0) standard errors

We highlight the main findings from these results. First, each of the pandemic-related disruptions are associated with increases in the average wait time, the average line length, 15 minute wait, and 30 minute wait to varying degrees. A PWS has the largest impact on average wait time and average line length, PPE has the second largest, and SD the third. Second, a PWS reduces the average time inside and average number of voters inside, since the utilization of check-in booths increases and voters are blocked from beginning step B within the voting process. Once inside, the utilization of the voting booths and optical ballot scanners are low, and voters can move through relatively quickly. Third, SD causes the largest increase to the amount of time spent inside and number of voters spent inside, since the number of voting booths is reduced. Fourth, Figures 2.5 – 2.7 highlight that the range of observed values increases for many voting metrics. Thus, given the same inputs, the probability of observing a value far from the mean is higher when compared to an election when no disruption is experienced. Election officials and voters must prepare for more unpredictability during a pandemic-disrupted election.

2.7 Case Study Results: Mitigating pandemic-related disruptions

We consider six mitigations to disruptions and quantify the benefits of each. These mitigations are to expand access to early voting, increase efforts to recruit poll workers, increase the number of optical ballot scanners, expand the physical footprint of polling locations, implement a priority queue for at-risk voters, and consolidate polling locations. We describe each mitigation in more detail in the subsections below. For each, we compare the outputs of the simulation when mitigating practices are put in place to an election without any mitigating practices but pandemic-related disruptions are experienced. Namely, PPW, SD, and PWS are all set to 1. Throughout this section, p-values are calculated using a paired t-test for the difference between the no-mitigation, pandemic-disruption election and an election with the various mitigating policies implemented unless otherwise noted. The baseline, no-mitigation inputs are indicated by "*" in each table, and baseline simulation outputs are presented in the first row of each table.

2.7.1 Expand access to early voting (EV)

As seen during the 2020 elections, there is a natural response by voters to avoid voting in-person on an Election Day during a public health emergency and instead use the absentee voting process. A Marquette poll conducted in May 2020 found that 54% of Wisconsin voters planned on voting early during the 2020 general election [73]. In the Wisconsin 2020 spring election, 74.4% of those who voted in Wisconsin voted early [218]. However, the access to absentee voting systems varies by jurisdiction and by state. As of May 2021, at least 12 states have moved to restrict access to mail based voting following the 2020 elections [19], which could reduce the number of people who can vote early. As a response to pandemic-related disruptions, state governments or election commissions may expand

access to absentee voting to improve in-person Election Day voting metrics and protect voter health. We quantify the impact of doing so.

Table 2.9: Mean value (standard deviation) of in-person voting metrics across 50 replications when experiencing various early voting rates.

Early Voting Rate	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
29.6%*	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
41.8%	57.5 (11.5)††	7.2 (0.1)††	0.60 (0.08)††	0.49 (0.08)††	49.6 (11.4)††	7.4 (0.3)††
54%	20.2 (5.8)††	6.9 (0.1)††	0.30 (0.07)††	0.21 (0.06)††	15.1 (4.9)††	5.8 (0.3)††
64.5%	5.3 (2.2)††	6.7 (0.1)††	0.10 (0.04)††	0.05 (0.02)††	3.3 (1.5)††	4.4 (0.3)††
75%	0.9 (0.5)††	6.6 (0.0)††	0.01 (0.01)††	0.00 (0.00)††	0.4 (0.2)††	3.1 (0.2)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than the baseline no-mitigation, pandemic-disrupted election at a level of 0.05 (0.01)

Table 2.9 presents the mean values and standard deviations of the in-person voting metrics across 50 replications for varying values of the early voting rate between 29.6% (baseline) – 75%. We find that increased early voting rates can substantially improve all in-person voting metrics, since the volume of voters served in-person on an Election Day is reduced. In the case of Milwaukee, WI, an early voting rate of 64.5% results in a mean average sojourn time (sum of wait time and time inside) of 11.95 ± 0.63 minutes. Even with an early voting rate of 54%, the voting metrics are substantially reduced from the pandemic-disrupted values shown in the first row of the table. When an early voting rate of 41.8% is experienced, many voting metrics are still high.

2.7.2 Increase poll worker recruitment (PWR)

Due to the COVID-19 pandemic, we evaluate the possibility of a shortage of poll workers that results in a reduced number of check-in booths. As a response to this, jurisdictions may increase efforts to recruit poll workers or request the National Guard to serve as poll workers. To support this decision, we quantify the impact of various policies to increase

the number of check-in booths. The pandemic-disrupted baseline is referred to as 100% – 1, since poll worker shortages reduce the number of check-in booths at each polling location by one. We consider three mitigation policies. The first is to add a check-in booth at 91 (half of 182) polling locations. We select the polling locations with the highest ratio of voting age population to check-in booths. We refer to this mitigating policy as "$50\% - 1$," since only half of the polling locations have one fewer check-in booth compared to the normal number. The second is to ensure all 182 polling locations retain their pre-pandemic number of check-in booths, referred to as "100%." The third is retain the normal number of check-in booths and add an additional check-in booth at the 91 (half of 182) polling locations with the highest ratio of voting age population, referred to as "$50\% + 1$." The "$50\% + 1$" may be considered when there is a surplus of potential poll workers as was experienced during the 2020 general election in many areas [171] and they are not needed in any other capacity. Instead of turning away potential poll workers, polling locations could increase the number of check-in booths.

Table 2.10: Mean value (standard deviation) of in-person voting metrics across 50 replications with varying numbers of check-in booths.

Check-in Booths	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
$100\% - 1$*	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
$50\% - 1$	44.4 (12.0)††	7.8 (0.1)††	0.69 (0.09)††	0.58 (0.10)††	53.3 (15.7)††	10.0 (0.5)††
100%	35.3 (9.9)††	8.0 (0.2)††	0.61 (0.09)††	0.49 (0.11)††	42.6 (13.3)††	10.3 (0.5)††
$50\% + 1$	24.9 (8.4)††	8.0 (0.2)††	0.53 (0.10)††	0.40 (0.12)††	30.7 (11.4)††	10.4 (0.6)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than the baseline no-mitigation, pandemic-disrupted election at a level of 0.05 (0.01)

Table 2.10 presents the mean values and standard deviations of the in-person voting system metrics across 50 replications for each of the policies. Increasing the number of check-in booths can be an effective mitigation to pandemic-related disruptions. The $50\% - 1$

mitigation significantly reduces the average wait time and the proportion of voters waiting 15 or 30 minutes or more to vote as compared to the baseline $100\% - 1$ case. However, this comes at a cost. The average time inside and average number of voters inside increases. As the number of check-in booths increases, the number of voting booths begins to constrain how quickly voters can move through the voting process. Adding more check-in booths also reduces the maximum capacity of the polling location, since space must be dedicated to the check-in booth and poll workers. The results suggest that increasing the number of check-in booths from the pandemic-disrupted level $(100\% - 1)$ has a marginally decreasing benefit to the voting metrics and that adding check-in booths cannot completely mitigate the impact of pandemic-related disruptions.

2.7.3 Number of optical ballot scanners (OBS)

Election officials may also consider increasing the number of optical ballot scanners to minimize the number of voters waiting in a queue inside to submit a ballot. The baseline number of optical ballot scanners is referred to as 100%, which represents a single scanner at each polling location. We consider the mitigation of adding an additional optical ballot scanner to each polling location (so that each has two), referred to as "$100\% + 1$." Adding this additional ballot scanner reduces the maximum capacity of the polling location by one, since there is less space for voters.

Table 2.11: Mean value (standard deviation) of in-person voting metrics across 50 replications when experiencing various number of optical ballot scanners.

Ballot Scanners	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
100%*	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
$100\% + 1$	115.5 (16.1)††	7.3 (0.1)††	0.82 (0.05)††	0.75 (0.06)††	108.7 (17.7)††	8.4 (0.3)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than no-mitigation, pandemic-disrupted election at a level of 0.05 (0.01)

Table 2.11 presents the mean values and standard deviations of the in-person voting system metrics across 50 replications for both levels of the number of optical ballot scanners. There is a statistically significant benefit of adding optical ballot scanners to the average time inside and average number of voters inside, although the benefits are small and may not be practically significant. However, adding an optical ballot scanner increases the average wait time, 15 minutes wait, 30 minute wait, and average line length, which may be counterintuitive. The worsening of voting metrics is a result of a reduced maximum capacity to allow for the space needed for the scanner.

2.7.4 Expand physical footprint (EPF)

To combat the impact of social distancing, polling locations may attempt to increase the footprint of the polling location, including the use of outdoor spaces. This would allow more individuals "within" the polling location and to retain a higher number of voting booths. To describe the impact of social distancing, we use the concept of a capacity factor, which is introduced in Sections 2.3.1 and 2.4.1. The pandemic-disrupted baseline is a capacity factor of 25% (relative to the pre-pandemic-capacity factor of 100%), which represents the implementation of a six-foot social distancing policy without increasing the footprint of the polling location. Increases to the capacity factor represents increases to the footprint of the polling locations while maintaining the six-foot radius. We study the impact of various levels of the capacity factor on six voter metrics.

Table 2.12 presents the mean values and standard deviations of the in-person voting system metrics across 50 replications for the considered levels of the capacity factor from 25% – 50%. Increasing the capacity factor from 25% to 33.3% impacts all metrics. The average wait time is reduced by 3.4 minutes, the time inside is reduced by 0.7 minutes, the proportion of voters waiting 15 minutes is reduced by 0.06, the proportion of voters waiting 30 minutes is reduced by 0.06, the average line length is reduced by 4.3 voters, and

Table 2.12: Mean value (standard deviation) of in-person voting metrics across 50 replications with various increases to the capacity factor.

Capacity Factor	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
25%*	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
33.3%	111.6 (14.9)††	6.8 (0.0)††	0.76 (0.04)††	0.69 (0.05)††	103.9 (16.1)††	7.8 (0.2)††
41.7%	111.5 (14.9)††	6.7 (0.0)††	0.76 (0.04)††	0.69 (0.05)††	103.8 (16.0)††	7.7 (0.2)††
50%	111.5 (14.9)††	6.6 (0.0)††	0.76 (0.04)††	0.69 (0.05)††	103.8 (16.0)††	7.7 (0.2)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than the baseline no-mitigation, pandemic-disrupted election at a level of 0.05 (0.01)

the average number of voters inside is reduced by 0.9 voter. The impact of increasing the capacity factor past 33.3% is minimal on any metric in the absence of other mitigations.

2.7.5 Queuing style (QS)

A FCFS queueing discipline is widely used in polling locations and is viewed as equitable [109]. However, a FCFS queueing discipline can lead to disparate health outcomes when voters have different health risks due to factors such as age. To address health concerns for subpopulations that are at heightened risk to the COVID-19 disease, election officials may consider implementing a self-identifying nonpreemptive *priority queue* (PQ) for high-risk voters in steps A, C, and E of the voting process. A priority queue would allow self-identifying high-risk voters to move to the front of the queue, but behind other high-risk voters already waiting. This reduces the time high-risk voters must wait, but also increases the wait time of low-risk voters. We examine a priority queue's impact on both high-risk and low-risk voters. In practice, a priority queue would likely be implemented by having a separate line for self-identifying high-risk voters. When voters self-identify, we assume no additional poll workers are needed to manage this system. Many polling locations already have a poll worker outside to manage lines and to inform voters of the necessary

documents, and these poll workers can help support the management of a priority queue if necessary.

Table 2.13 presents the mean values and standard deviations of the in-person voting metrics across 50 replications for voting systems that use the traditional FCFS queuing policy and those that implement a priority queuing policy. Table 2.14 presents the mean and standard deviation of the average wait time and average sojourn time experienced by low-risk and high-risk sub-populations.

Table 2.13: Mean value (standard deviation) of in-person voting metrics across 50 replications with first-come, first served (FCFS) or priority queue (PQ) queuing styles.

Queuing Style	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
FCFS *	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
PQ	115.0 (16.0)	7.5 (0.1)	0.72 (0.04)††	0.67 (0.05)††	108.1 (17.6)	8.7 (0.3)

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than the baseline no-mitigation, pandemic-disrupted election at a level of 0.05 (0.01)

Table 2.14: Mean value (standard deviation) of average wait time and average sojourn time for low- and high-risk voters across 50 replications for first-come, first-served (FCFS) and priority queue (PQ) queueing policies.

	Low-risk Voters		High-risk Voters	
Queuing Style	Avg. Wait Time	Avg. Sojourn Time	Avg. Wait Time	Avg. Sojourn Time
FCFS *	115.0 (16.0)	122.6 (16.2)	115.1 (16.1)	122.5 (16.1)
PQ	133.3 (18.6)††	140.9 (18.7)††	0.8 (0.0)††	7.6 (0.1)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than the baseline no-mitigation, pandemic-disrupted election at level $p < 0.05$ ($p < 0.01$)

Implementing a priority queue policy reduces the mean proportion of voters waiting at least 15 minutes (from 0.82 to 0.72) and at least 30 minutes (from 0.75 to 0.67). This is

because high-risk voters who experience a long wait time when a FCFS queueing policy is used are served more quickly when a priority queueing policy is used. High-risk voters wait 0.8 minutes on average while low-risk voters wait 133.3 minutes when a priority queue is implemented. High-risk voters also spend less time inside when priority queues are implemented (6.8 minutes) versus FCFS queues (7.5 minutes). However, implementing a priority queue requires consideration of the impact of queuing policy on all components of the election system by election officials, not just the process based metrics analyzed within this chapter. If a priority queue is implemented, election officials should take care to address the psychological injustice associated with priority queues [109].

2.7.6 Consolidate Polling Locations (CPL)

In response to poll worker shortages and to manage the complexity of in-person voting during a pandemic, election commissions may consider consolidating polling locations as was done in Milwaukee for the 2020 spring election [28]. To study the impact of such a decision, we employ the polling location consolidation scheme that was used for the 2020 spring election. Other consolidation schemes may be employed depending on the circumstance. However, selecting which polling locations should remain and how to assign wards to polling locations are decisions that requires extensive study and falls outside the scope of this chapter. The approach used in this chapter provides an estimate on the impact of any consolidation scheme and provides insights for election planning.

We compare the traditional election system with 182 polling locations to four different mitigating policies. The first is to reduce the number of polling location to five and retain the same total number of check-in booths and ballot scanners within the system. We redistribute the resources to the five polling locations according to how wards were assigned to polling locations in the 2020 spring election. We refer to this policy as '5 (100%)' as we have 5 polling locations and 100% of the resources retained. However, this results in an average of

101.6 check-in booths and 36.4 ballot scanners located at each of the five polling locations. This is unrealistic due to space limitations and managerial complexities. Additionally, having enough poll workers to staff all of the check-in booths would eliminate much of the need to consolidate polling locations. Therefore, the other three mitigating policies represent voting systems with a reduced number of check-in booths and ballot scanners. The first of these is '5 (80%)', which represents a voting system with 5 polling locations and 80% of the number of check-in booths and ballot scanners. This is, on average, 81.8 check-in booths and 29.4 ballot scanners. The second is '5 (60%)', which represents a voting system with 5 polling locations and 60% of the number of check-in booths and ballot scanners(an average of 61.4 check-in booths and 22.2 ballot scanners). The last is '5 (40%)', which represents a voting system with 5 polling locations and 40% of the number of check-in booths and ballot scanners (an average of 41.2 check-in booths and 15.0 ballot scanners). In each case, we recalculate the number of voting booths and the capacity of the consolidated polling locations according to the formulas described in Section 2.4. Table 2.15 presents the mean values and standard deviations of the in-person voting system metrics across 50 replications under each mitigating policy.

Table 2.15: Mean value (standard deviation) of in-person voting metrics across 50 replications with various polling location consolidation schemes.

Polling Locations	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
182*	115.0 (16.0)	7.5 (0.1)	0.82 (0.05)	0.75 (0.06)	108.2 (17.6)	8.7 (0.3)
5 (100%)	66.8 (16.7)††	6.7 (0.1)††	0.86 (0.06)††	0.78 (0.09)††	2813.1 (737.6)††	305.1 (7.2)††
5 (80%)††	159.7 (21.9)††	6.6 (0.0)††	0.97 (0.01)††	0.94 (0.01)††	5558.8 (774.1)††	248.6 (0.9)††
5 (60%)	321.6 (28.0)††	6.6 (0.0)††	1.0 (0.0)††	1.0 (0.0)††	8490.9 (748.3)††	187.8 (0.5)††
5 (40%)	640.2 (40.2)††	6.6 (0.0)††	1.0 (0.0)††	1.0 (0.0)††	11430.7 (724.1)††	126.8 (0.2)††

*Baseline no-mitigation, pandemic-disrupted election
† (††) Statistically different than no-mitigation, pandemic-disrupted election at level 0.05 (0.01)

We find that consolidating polling location can improve the average wait time and the average time inside if the same number of check-in booths and ballot scanners can be

retained within the voting system (the 5 (100%) case). However this comes at a cost in an increase in the average line length and the average number of voters inside, which average 2813.1 and 305.1 in the 5 (100%) case. The former can discourage voters from entering the line to vote at a polling location. The latter can increase the likelihood of disease spread, depending on the infectious disease under consideration and the layout at the polling location.

When the number of check in booths and ballot scanners must be reduced (the 5 (80%), 5 (60%), and 5 (40%) cases), the average wait time, 15 minute wait, 30 minute wait, and average line length substantially increase to levels that would not be sustainable during an election. With 80% of the resources, the average wait time increases to over two and a half hours, and with 60% of the resources, the average wait time increases to over five hours.

These results indicate that there are major drawbacks to consolidating polling locations when the number of check-in booths and ballot scanners must be reduced. In addition, consolidating polling locations increases the distance a voter must travel to a polling location in nearly all cases, which can suppress voter turnout. We estimate the average geodesic distance of voters from their assigned polling location when there are 182 or 5 polling locations used by using random voter locations within each ward. We estimate a voting age individual in Milwaukee to be 0.38 miles from their polling location when there are 182 polling locations and 2.16 miles from their consolidated polling location when there are 5 polling locations. The maximum distance a voter is from their polling location is 3.04 miles and 7.61 miles for the 182 and 5 polling locations, respectively. Research indicates that the likelihood a voter casts a ballot decreases with the distance they must travel to a polling location [25]. Initial research from the 2020 spring election in Milwaukee suggests the voter turnout decreased by 8.5% overall and by 10.2% among the Black voting population due to the consolidation of polling locations [134]. Moreover, consolidating polling locations requires voters from many locations to congregate in a single, centralized location. This

increases the likelihood of inter-community disease spread while also reducing the ability for public health officials to perform contract tracing.

Avoiding the consolidation of polling locations may be particularly difficult when election officials are unable to recruit poll workers, since there are often legal requirements for the number of election inspectors at each polling location.

2.7.7 Optimizing the mitigation portfolio

Thus far, we have focused on the impact of the each mitigation independently. However, two mitigations may address the same concern in an election process and, others may be symbiotic and further improve the performance of the in-person voting process. We create a full factorial design for the four mitigations we found to be reasonable minimally viable mitigations to pandemic-related disruptions based on the analysis in Section 2.7.1 – 2.7.6. The first, EV, is to expand access to early voting so that the voter turnout rate increases to to 54% (from 29.6%). The second, PWR, is to devote effort to poll worker recruitment so that fewer locations lose a check-in booth (50% – 1) (instead of all polling locations losing a check-in booth). This means that the 91 polling locations with the highest ratio of voting age individuals to check-in booths retains its normal (pre-pandemic) level. The third, EPF, is to expand the physical footprint so the capacity factor at all polling locations is increased to 33.3%. Lastly, QS, is to change the queueing policy from FCFS to PQ throughout the voting process. Table 2.16 summarizes the mitigations, the parameter it changes, and levels considered within the design.

We use linear regression to analyze the simulation outputs. Table 2.17 presents the coefficients and associated p-values for each first and higher order term in the full factorial design. The row for the variable 'Constant' indicates the output from the simulation during an election where all disruptions are experienced and no mitigations put in place, i.e., all IDs are set to zero. Within this table, if the coefficient is negative, then the metric value

Table 2.16: Mitigation Variable Values.

			Parameter Value	
Mitigation	ID	Impacted Parameter	ID = 0	ID = 1
Expand early voting	EV	Early voting rate	29.6%	54%
Poll worker recruitment	PWR	Check-in Booths	100%-1	50%-1
Expand physical Footprint	EPF	Capacity factor	25%	33.3%
Queue Style	QS	Queuing style	FCFS	PQ

is expected to improve with the mitigating policy. For example, poll worker recruitment (PWR = 1) decreases the average wait time by 70.6 minutes. Higher order terms (e.g., PWR*EV) capture the interaction effects when both mitigations are implemented.

Table 2.17: Ordinary least squares regression coefficients (p-values) for each voting metric when various mitigations are implemented (ID=1) or not (ID=0). Results are presented for an expected voter turnout of 57.2%.

	Coefficient (p-value*)					
Variable	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant	115.0 (0.00)	7.5 (0.00)	0.82 (0.00)	0.75 (0.00)	108.2 (0.00)	8.7 (0.00)
EV	-94.9 (0.00)	-0.6 (0.00)	-0.52 (0.00)	-0.54 (0.00)	-93.1 (0.00)	-2.8 (0.00)
PWR	-70.6 (0.00)	0.3 (0.00)	-0.13 (0.00)	-0.17 (0.00)	-54.9 (0.00)	1.3 (0.00)
EPF	-3.5 (0.26)	-0.7 (0.00)	-0.06 (0.00)	-0.06 (0.00)	-4.2 (0.20)	-0.8 (0.00)
QS	-0.0 (1.00)	0.0 (0.99)	-0.10 (0.00)	-0.08 (0.00)	-0.0 (1.00)	0.0 (0.99)
PWR*EV	53.4 (0.00)	-0.2 (0.00)	-0.11 (0.00)	-0.03 (0.13)	42.3 (0.00)	-1.2 (0.00)
PWR*EPF	-2.4 (0.52)	-0.3 (0.00)	-0.03 (0.08)	-0.04 (0.10)	-2.9 (0.51)	-0.3 (0.00)
PWR*QS	0.0 (0.99)	-0.0 (1.00)	0.02 (0.24)	0.03 (0.15)	0.0 (1.00)	0.0 (1.00)
EV*EPF	3.3 (0.31)	0.5 (0.00)	0.05 (0.00)	0.06 (0.00)	4.1 (0.24)	0.6 (0.00)
EV*QS	0.0 (1.00)	-0.0 (0.99)	0.08 (0.00)	0.07 (0.00)	0.0 (1.00)	-0.0 (1.00)
EPF*QS	-0.0 (1.00)	0.0 (0.99)	0.01 (0.48)	0.01 (0.65)	-0.0 (1.00)	0.0 (0.99)
PWR*EV*EPF	2.3 (0.56)	0.2 (0.00)	0.03 (0.25)	0.04 (0.15)	2.8 (0.53)	0.3 (0.04)
PWR*EV*QS	-0.0 (1.00)	0.0 (1.00)	0.01 (0.71)	-0.02 (0.48)	-0.0 (1.00)	0.0 (1.00)
PWR*EPF*QS	-0.0 (1.00)	0.0 (0.99)	0.00 (0.84)	0.00 (0.86)	-0.0 (1.00)	0.0 (1.00)
EV*EPF*QS	0.0 (1.00)	-0.0 (0.99)	-0.01 (0.69)	-0.01 (0.75)	0.0 (1.00)	-0.0 (1.00)
PWR*EV*EPF*QS	0.0 (1.00)	-0.0 (1.00)	-0.01 (0.88)	-0.01 (0.88)	0.0 (1.00)	-0.0 (1.00)

*Using heteroscedasticity robust (HC0) standard errors

Tables 2.18 and 2.19 present the regression coefficients and p-values for mitigations

with voter turnout rates of 47.2% and 67.2%, respectively. These are interpreted the same as Table 2.17 within the main text. When voter turnout is 47.2%, the impact is of mitigating policies is less than when the voter turnout rate is 57.2%. However, the relative magnitude of primary and interaction effects are the similar to when the voter turnout rate is 57.2%. Conversely, when the voter turnout rate is 67.2%, the impact of mitigating policies is larger than when the voter turnout rate is 57.2%.

Table 2.18: Ordinary least squares regression coefficients (p-values) for each voting metric when various mitigations are implemented (ID=1) or not (ID=0). Results are presented for an expected voter turnout of 47.2%.

Variable	Coefficient (p-value*)					
	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant	57.0 (0.00)	7.2 (0.00)	0.60 (0.00)	0.49 (0.00)	49.2 (0.00)	7.4 (0.00)
CB	-42.9 (0.00)	0.2 (0.00)	-0.25 (0.00)	-0.29 (0.00)	-34.7 (0.00)	0.5 (0.00)
EV	-49.5 (0.00)	-0.4 (0.00)	-0.46 (0.00)	-0.41 (0.00)	-44.3 (0.00)	-2.6 (0.00)
EPF	-1.0 (0.65)	-0.5 (0.00)	-0.03 (0.03)	-0.01 (0.32)	-1.0 (0.64)	-0.5 (0.00)
QS	-0.0 (1.00)	0.0 (0.99)	-0.07 (0.00)	-0.04 (0.01)	-0.0 (1.00)	0.0 (1.00)
CB*EV	36.2 (0.00)	-0.2 (0.00)	0.12 (0.00)	0.22 (0.00)	30.4 (0.00)	-0.5 (0.00)
CB*EPF	-0.7 (0.76)	-0.2 (0.00)	-0.02 (0.35)	-0.01 (0.60)	-0.8 (0.76)	-0.2 (0.07)
CB*QS	0.0 (1.00)	-0.0 (1.00)	0.04 (0.12)	0.05 (0.06)	0.0 (1.00)	-0.0 (1.00)
EV*EPF	1.0 (0.67)	0.4 (0.00)	0.03 (0.08)	0.01 (0.36)	1.0 (0.65)	0.4 (0.00)
EV*QS	0.0 (1.00)	-0.0 (0.99)	0.06 (0.00)	0.04 (0.01)	0.0 (1.00)	-0.0 (1.00)
EPF*QS	0.0 (1.00)	-0.0 (0.99)	0.00 (0.89)	-0.00 (0.92)	0.0 (1.00)	-0.0 (1.00)
CB*EV*EPF	0.7 (0.77)	0.1 (0.00)	0.02 (0.40)	0.01 (0.61)	0.7 (0.77)	0.1 (0.22)
CB*EV*QS	-0.0 (1.00)	-0.0 (1.00)	-0.03 (0.27)	-0.05 (0.07)	-0.0 (1.00)	-0.0 (1.00)
CB*EPF*QS	0.0 (1.00)	0.0 (0.99)	0.00 (0.94)	-0.00 (0.97)	0.0 (1.00)	0.0 (1.00)
EV*EPF*QS	-0.0 (1.00)	0.0 (1.00)	-0.00 (0.91)	0.00 (0.93)	-0.0 (1.00)	0.0 (1.00)
CB*EV*EPF*QS	-0.0 (1.00)	-0.0 (0.99)	-0.00 (0.95)	0.00 (0.97)	-0.0 (1.00)	-0.0 (1.00)

*Using heteroscedasticity robust (HC0) standard errors

We find that when all mitigations are implemented, we expect the lowest values for the average wait time, 15 minute wait, 30 minute wait, and average line length. Interestingly, this is not the case for the average time inside and average number of voters inside. We find the average time inside and average voters inside is lowest when a high early voting rate is experienced (EV = 1) and polling locations expand their physical footprint (EPF =

Table 2.19: Ordinary least squares regression coefficients (p-values) for each voting metric when various mitigations are implemented (ID=1) or not (ID=0). Results are presented for an expected voter turnout of 67.2%.

Variable	Coefficient (p-value*)					
	Avg. Wait Time	Avg. Time Inside	15 Minute Wait	30 Minute Wait	Avg. Line Length	Avg. Voters Inside
Constant const	185.3 (0.00)	7.7 (0.00)	0.94 (0.00)	0.90 (0.00)	183.9 (0.00)	9.5 (0.00)
CB	-87.7 (0.00)	0.4 (0.00)	-0.03 (0.00)	-0.04 (0.00)	-60.0 (0.00)	1.8 (0.00)
EV	-143.1 (0.00)	-0.6 (0.00)	-0.44 (0.00)	-0.51 (0.00)	-149.0 (0.00)	-2.6 (0.00)
EPF	-8.5 (0.02)	-0.9 (0.00)	-0.07 (0.00)	-0.08 (0.00)	-11.0 (0.01)	-1.1 (0.00)
QS	0.0 (1.00)	0.0 (0.93)	-0.12 (0.00)	-0.11 (0.00)	0.0 (1.00)	0.0 (0.97)
CB*EV	54.5 (0.00)	-0.3 (0.00)	-0.23 (0.00)	-0.25 (0.00)	33.6 (0.00)	-1.5 (0.00)
CB*EPF	-5.6 (0.25)	-0.3 (0.00)	-0.03 (0.00)	-0.04 (0.00)	-7.0 (0.24)	-0.4 (0.00)
CB*QS	0.0 (0.99)	-0.0 (0.98)	0.01 (0.49)	0.01 (0.39)	0.0 (1.00)	-0.0 (0.99)
EV*EPF	7.9 (0.06)	0.5 (0.00)	0.05 (0.00)	0.07 (0.00)	10.5 (0.02)	0.7 (0.00)
EV*QS	-0.0 (1.00)	-0.0 (0.97)	0.07 (0.00)	0.08 (0.00)	-0.0 (1.00)	-0.0 (0.99)
EPF*QS	-0.0 (1.00)	-0.0 (0.95)	0.01 (0.13)	0.01 (0.15)	-0.0 (1.00)	-0.0 (0.99)
CB*EV*EPF	5.1 (0.32)	0.2 (0.00)	0.02 (0.45)	0.04 (0.11)	6.6 (0.30)	0.2 (0.02)
CB*EV*QS	-0.0 (1.00)	0.0 (0.98)	0.03 (0.18)	0.03 (0.18)	-0.0 (1.00)	0.0 (0.99)
CB*EPF*QS	-0.0 (1.00)	0.0 (0.97)	0.01 (0.65)	0.01 (0.65)	-0.0 (1.00)	0.0 (0.99)
EV*EPF*QS	0.0 (1.00)	0.0 (0.98)	-0.01 (0.65)	-0.01 (0.50)	0.0 (1.00)	0.0 (0.99)
CB*EV*EPF*QS	0.0 (1.00)	-0.0 (0.99)	-0.00 (0.89)	-0.01 (0.79)	0.0 (1.00)	-0.0 (0.99)

*Using heteroscedasticity robust (HC0) standard errors

1), but no other mitigations are put in place. These mitigations reduce congestion inside a polling location, since there are fewer voters voting in person and there is an increased number of voting booths. We also find that when there is a high early voting rate (EV = 1), the impact of poll worker recruitment (PWR = 1) on the average wait time and average line length is greatly reduced (and vice versa). However, the mitigations together are mutually beneficial for the average time inside, 15 minute wait, the 30 minute wait, and average voters inside, as shown by negative values of the the interaction terms. We find the queuing style has no significant impact on the average metrics. These results suggests there are multiple options that can mitigate the effect of pandemic-related disruptions on polling locations depending on the resources and flexibility designated to election administrators.

2.8 Discussion

The COVID-19 pandemic has highlighted the need to mitigate disruptions to in-person voting systems. Elections held during the COVID-19 pandemic motivate consideration of new alternatives for the design and operation of in-person voting systems to ensure that voting systems are resilient to poll worker shortages, social distancing, and the use of PPE and sanitation and to mitigate the risk of infectious disease transmission. Rigorous planning and analysis using analytical methods such as discrete event simulation can help election officials mitigate disruptions by prioritizing planning efforts. We provide recommendations for election officials preparing for an election using a case study of Milwaukee, WI.

As with any model of a system, there are dynamics of the voting process that are not captured in great detail, and there are several limitations of the analysis presented in this chapter. First and foremost, this simulation and case study focuses on the queueing dynamics of the voting process. Other technical and social aspects should be considered when designing a voting system. It is likely that changes and improvements to the voting process may influence voter choices, which we do not consider. For example, mitigating efforts to reduce voter wait times and the distances voters must travel to polling locations may result in higher voter turnout, since the "cost" of casting a vote is lower [22]. Our analysis does not explicitly account for the impact of long lines on voter turnout. Individuals may see or hear about line lengths at their local polling location and choose not to vote. Due to a pandemic, voters may be required to wait outside before checking-in to reduce the viral load inside, even if there is room within the polling location. Social distancing in the check-in queue may give the appearance of longer lines compared to elections that occurred before the pandemic. The appearance of long lines may discourage participation in the voting process, since voters may (incorrectly) assume the cost to vote is high [22, 109]. We

do not consider curbside voting, which may increase the utilization of existing volunteer resources or require dedicated volunteers. We also assume that voters' use of PPE does not impact the time to mark a ballot. It is possible that voting times may increase slightly due to mask wearing (e.g., fogging of eyeglasses). Moreover, depending on the election, the number of items to vote on can change the length of time it takes to mark a ballot. Additional analysis not reported in this chapter suggests the impact of this is small. However, when the maximum capacity and number of voting booths is reduced, the impact of increased voting times is larger.

The simulation analysis uses real-world data to estimate realistic input parameters; however, most of the data is based on elections that occurred before the COVID-19 pandemic and some were not collected in Milwaukee, WI. It is possible that some of the inputs may not reflect actual conditions that would be experienced on any Election Day. For example, we do not consider a change in voter arrival pattern compared to previous elections. With more voters working from home during a pandemic, this pattern may change. If the pattern becomes more evenly distributed throughout the day, various metrics associated with wait times and line lengths would improve. An increase in early voting rates may also shift the demographics of in-person voters on an Election Day. This may change the time to complete each step of the voting process. An analysis of poll conditions is needed to evaluate whether there were changes to voter arrival patterns and demographics. The number of individuals who would self-identify as high-risk is also unknown before an election and likely varies between polling locations due to socioeconomic status and health of the community. With more individuals self-identifying as high-risk, low-risk voters must wait longer while high-risk voters go through the voting process. Additionally, when the lines are very long, some dedicated poll workers may be needed to ensure high-risk voters are aware of the priority queue. Finally, the simulation model does not explicitly consider check-in booth or ballot scanner downtime. When resource downtime is experienced, this

introduces additional delays that lead to longer wait times and line lengths. A system with more polling locations experiences a larger impact from resource downtime [21].

Planning for an election is a topic of national concern, with elections considered to be part of our nation's critical infrastructure. This chapter presents a detailed analysis of a discrete event simulation model that supports planning for elections. The results can be used to inform election planning decisions and help election officials operate elections that are efficient, equitable, accessible, and safe.

Chapter 3

Optimal consolidation of polling locations

3.1 Introduction

Election planning is a critical step to creating effective and resilient voting systems. The ultimate goal of election planning is to design voting systems that are easy to use, result in high voter participation, and are robust to a number of election conditions. Three of the most important planning decisions made by election officials are deciding where to locate polling locations, how to assign voters to polling locations, and where to allocate critical voting resources (e.g., voting machines). Each of these decisions influence the participation of voters during an election [26, 188] and the effectiveness of the voting system.

Typically, polling locations and the assignment of voters to polling locations remain constant across multiple elections to avoid causing confusion among voters. However, buildings typically used as a polling location, such as churches or senior centers, may no longer exist or may no longer be used[1] [211]. Ideally, election officials can identify a new

[1]This was prevalent during the COVID-19 pandemic due to health concerns.

building near the former polling location at which the impacted individuals can vote when this occurs so that the voting system is minimally disrupted. However, there are often a limited number of potential polling locations due to legal requirements (e.g., Americans with Disabilities Act). When no other options exist, the individuals typically assigned to vote at two or more unique polling locations must instead vote at the same location. This practice is known as polling location consolidation.

Many logistical and financial challenges also cause election officials to consolidate polling locations [211]. In some areas, growing populations can render a previously used polling location too small. On the contrary, the use of non-traditional voting methods has steadily grown in many jurisdictions since 1996 [165]. In 2020, 69.4% of voters cast their ballot either absentee or in-person prior to Election Day [165]. As a result, the cost of managing non-traditional voting pathways has increased and the utilization of the in-person voting system on Election Day has decreased. Election officials may consolidate polling locations to reduce the cost of facilitating in-person voting, especially during non-presidential or primary elections when voter turnout is typically low [211]. In addition, election officials rely on critical voting resources (e.g., poll workers, voting machines) to facilitate an election. A disruption (e.g., pandemic, cyber-physical attack) to the voting system may reduce the number of voting resources available, and election officials may consolidate polling locations to more effectively utilize the resources in response [211]. This course of action was taken in many jurisdictions during the COVID-19 pandemic due to poll worker shortages.

While sometimes necessary, the decision to consolidate polling locations cannot be taken lightly. Empirical research suggests that prior consolidation attempts reduced voter turnout by 1.85 to 8.7 percent [18, 123, 135]. This is largely because voters must travel further to their polling locations, which has been shown to decrease voter participation [61, 26]. However, the planning decisions must be made well before the actual conditions

on Election Day are known, and the decision not to consolidate may ultimately cause voters to wait in line a substantial amount of time to cast a ballot because there is an insufficient number of voting resources. This is undesirable, since long wait times have also been shown to reduce voter participation and have been shown to reduce voters' trust in the election process [188, 92]. The importance of reducing waiting during the voting process is highlighted by the goal set by the Presidential Commission on Election Administration that no voter wait longer than 30 minutes to vote [150].

When appropriate, deciding how to consolidate polling locations effectively is challenging. In many cases, the allocation of voting resources and the consolidation of polling locations are interdependent decisions. When the supply of voting resources is limited, there may not be a sufficient number of voting resources that can be allocated to a polling location to ensure voters do not wait long to cast a ballot. This suggests that the polling location should be consolidated with another that can accommodate the additional voters, but deciding which location is non-trivial. Consolidation schemes identified using simple rules-of-thumb that minimize the distance voters must travel to their new polling location, such as consolidating the closest polling locations, may not be feasible, since each polling location has limited physical space for voters and voting resources.

We posit that analytical techniques can help election officials make consolidation decision so that the impact on voters is minimal. In this chapter, we formalize the constraints and criteria election officials should follow when making consolidation decisions as the polling location consolidation problem (PLCP). We then introduce an integer program (IP) of the PLCP to determine when and how to consolidate polling locations to best achieve election goals. The PLCP simultaneously determines which polling locations to use in the upcoming election, how to assign voters to polling locations, and where to allocate critical voting resources. The objective of the PLCP is to minimize the additional time or distance voters spend traveling to their updated polling locations. In this way, the PLCP minimizes

the burden that consolidation places on voters. The PLCP also considers the amount of time voters spend waiting to vote once arriving to their polling location, since waiting can disenfranchise voters. Embedded within the PLCP is a chance-constraint to ensure most voters wait no longer than 30 minutes to vote. To formulate the chance constraint, we model each polling location as a $M/M/c$ queueing system. However, the number of servers, c, available at each polling location is determined by the resource allocation decisions, and therefore the wait time experienced by voters is endogenous to the PLCP.

The PCLP can support election officials in developing pre-vetted contingency plans that allow election officials to be responsive to changing election conditions. We study the theoretical properties of the PLCP and show that it is NP-Hard to identify feasible, let alone optimal, consolidation schemes. This highlights the complexity of decisions currently being made by election officials and motivates the use of optimization for this problem. We then study the solutions identified by the PLCP using a detailed case study of Richland County, South Carolina. The case study highlights a number of implications for practice. First, consolidating a moderate number of polling locations can be an effective mitigation to certain election conditions, such as voting resource shortages. However, consolidating so that just a small number of polling locations remain can substantially increase the distance that voters must travel to their polling location and is likely undesirable. We find that the burden placed on voters by consolidation can be substantially reduced by using large, strategically located buildings as polling locations during consolidation.

The remainder of the chapter is structured as follows. In Section 3.2, we review the literature related to polling location consolidation. In Section 3.3, we formalize the PLCP, and we study the properties of the model in Section 3.4. In Section 3.5, we introduce a case study of Richland County, South Carolina and highlight implications for practice. Finally, in Section 3.6, we summarize the policy implications highlighted within our case study and discuss future research opportunities.

3.2 Literature review

To the best of our knowledge, this chapter represents the first operations research approach to consolidating polling locations. The problem we consider in this chapter combines concepts from facility location, queueing, and districting to identify consolidation schemes that reduce the disruption to the voting system while ensuring that voters do not wait too long to vote on Election Day.

The PLCP is closely related to facility location problems. In facility location problems, the placement of facilities must be selected in a way that best serves a collection of demands. Often distance is used as a measure of a facility's ability to serve a demand. The literature on facility location problems is extensive in terms of modeling approach and application [5, 225, 17, 68, 176]. The three primary facility location problems are the maximal covering location problem [36], the location set covering problem [197], and the p-median problem [86]. In the the maximal covering problem, a set number of facilities are located such that the maximal number of demands are *covered*, or within some predefined distance, by at least one selected location. In the location set covering problem, a minimal number facility locations are selected while all demands must be covered by a facility. In the p-median problem, p facilities are located such that the sum of weighted distances from the demands to the facility closest to them is minimized. In many applications, a facility may be capacitated in a way that limits the number demands it can serve. In some problems, limited facility capacity leads to congestion and queueing. Marianov and Serra [118] study the problem of locating a minimal number of facilities and allocating a minimum number of resources to the facilities such that they can provide "good" service to demands. To do so, they model each facility as a $M/M/c$ queue and introduce a chance constraint to capture queueing dynamics. They consider two measures of good service, line length and wait time, and introduce a model for each.

The importance of queueing theory to in-person voting systems is well-demonstrated in the literature [189]. Voting queues increase the "cost" of voting, disenfranchise voters, and decrease trust in the election system [189]. Following the 2012 presidential election, the Presidential Commission on Election Administration set a goal that no voter waits more than 30 minutes to vote [150], which increased the importance of queueing models for voting systems. Queueing theory has been used to study why lines form and study how lines can be mitigated [189]. Among the chief causes of voting queues is a mismatch between the number of voters assigned to a polling location and the number of voting resources allocated to the polling location [189]. In many voting processes, voting machines have been recognized as a bottleneck in the voting process [221], and a stream of research has attempted to identify best practices of allocating voting machines to polling locations [6, 221, 112, 214, 224]. Similarly, other research has focused on the allocation of staff and voting booths (rather than voting machines) to polling locations [105]. More recent research has investigated the allocation of space, poll workers, voting booths, and optical ballot scanners due to problems that arose during the COVID-19 pandemic [166]. However, the selection of polling locations and the assignment of voters to polling locations are assumed to be static within most models.

The problem of locating polling locations to best serve voters is a critical, yet relatively understudied, challenge. The impact of changing or moving polling locations has been shown to be substantial; even a small increase in the distance between a voter and their assigned polling location can significantly reduce voter turnout [88, 26]. It has also been shown that polling locations can be nefariously selected to change the outcome of an election [72]. Thus, it is critical to have transparent, analytical methods that support election administrators in selecting polling locations. However, to the best of our knowledge, there are only four research papers that study this type of problem and all focus on voting systems outside of the United States. Table 3.1 summarizes the problem dynamics considered in

each of the four papers and compares them to this chapter. Ghiani et al. [79] introduce the capacitated plant location problem with multiple facilities in the same site (CPLPM). They apply the CPLPM to the problem of locating polling stations in an Italian municipality. Murata and Konishi [138] study a evolutionary multiobjective optimization algorithm to select polling locations while increasing turnout in Japan. Durán et al. [60] study how to reassign voters to polling locations to minimize the total distance traveled to the assigned polling locations in Argentina while requiring that each polling location serve the same number of voters. They extend their model to redistribute polling stations among the polling locations to minimize the sum of travel and wait times. Kim [104] presents the capacitated double p-median problem with preference (CDPMP-P) to simultaneously locate heterogeneously preferred polling locations and delimit precinct boundaries in Korea. They require that the number of assigned voters to the polling locations is balanced. They explicitly enforce that the assignment of voters to polling locations creates contiguous districts. Contiguity is enforced within their model using flow-based constraints.

Table 3.1: Comparison of problem dynamics considered by existing polling location selection literature.

Paper	Polling Location Selection	Capacitated Faciliities	Resource Allocation	Contiguous Districts	Endogenous Voter Wait time
Ghiani et al. [79]	✓	✓			
Murata and Konishi [138]	✓				
Durán et al. [60]		✓	✓		✓
Kim [104]	✓			✓	
This chapter	✓	✓	✓	✓	✓

The problem of constructing contiguous districts is common in many political problems, particularly political redistricting. Redistricting problems focus on designing congressional districts that meet a specific set of requirements (e.g., population balance, contiguity, and compactness) [158]. Most models aim to avoid gerrymandering, or the practice of designing districts in a way to favor a specific political party. A chief modeling challenge of

these models is the need to ensure contiguity of the districts [205, 192]. Hess offered the first optimization-based approach to perform political redistricting [91]. While their objective seeks compact districts, contiguity is not enforced. Later research explicitly enforced contiguity through the use of constraints [169, 143, 205]. Validi et al. [205] reviews two models to impose contiguity proposed by [143] and propose two new models.

3.3 Problem Definition

Voting systems are typically planned and managed by county or municipality election officials with limited time and resources. One component of the voting system is the *in-person* voting system where voters physically vote at a designated polling locations. When designing the in-person voting system, election officials must use multiple key voting metrics to inform interdependent decisions. The two primary metrics are the distance between voters and their designated polling locations and the time that voters ultimately wait once arriving there to vote, since both impact voter participation [26, 188]. In addition, election officials must design the voting system in a way that is affordable given their budget.

Within this section, we explicitly consider three interrelated consolidation decisions made by election officials. The first is to determine how many polling locations to use in the upcoming election and where to locate polling locations. The second is how to assign voters to polling locations. The third is where to distribute critical voting resources to ensure the voting process is quick and reliable. These decisions determine if polling location consolidation occurs and how the voting system is reconstructed if so.

Within this section, we formalize these decisions as the polling location consolidation problem (PLCP) and then introduce an integer program to support election officials in determining when and how to consolidate polling locations in a way that ensures key voter

metrics are met. The notation used throughout this section is presented in Table 3.2.

Table 3.2: Problem Notation

Sets		
I	=	population districts
J	=	potential polling locations
$\hat{J} \subset J$	=	polling locations that cannot be active
$N_i \subseteq I$	=	population districts neighboring $i \in I$
R	=	non-server resources

Parameters		
$j(i)$	=	the standard polling location of population district i
$i(j)$	=	the population district in which polling location $j \in J$ is located
d_{ij}	=	the distance between population district $i \in I$ and polling location $j \in J$
a_{rij}	=	the number of resource type $r \in R$ demanded by population district $i \in I$ if assigned to polling location $j \in J$
$\bar{a}_{rj}$	=	the resource capacity of type $r \in R$ at polling location $j \in J$
c_j	=	the maximum number of server resources that can be located at location $j \in J$
$s_{\max}$	=	the global supply of server resources available to the jurisdiction
n	=	the maximum number of polling locations to be active
λ_i	=	that arrival rate of voters from location $i \in I$ to the voting system, $\lambda_i > 0$
μ_j	=	the service rate of the server resource (e.g., check-in booths) at $j \in J$
α	=	acceptable proportion of voters that wait more than τ minutes at polling location $j \in J$
$\lambda_{\alpha m}$	=	the maximal voter arrival rate at a polling location $j \in J$ with $m \in \{1, 2, \ldots, c_j\}$ server resources, $\lambda_{\alpha m} > 0$
p_i	=	the number of voters in population district $i \in I$, $p_i > 0$

Decision Variables		
x_{ij}	=	1 if population district $i \in I$ is assigned to polling location $j \in J$, zero otherwise
y_{jm}	=	1 if at least m server resources are allocated to location $j \in J$, $m = 0, 1, 2, \ldots c_j$, zero otherwise. Note that y_{j1} indicates if the polling location is active.

3.3.1 The Polling Location Consolidation Problem (PLCP)

The in-person voting system is defined by two primary components. The first is a collection of existing polling locations and buildings that could potentially serve as polling locations,

denoted J. When a location in J is used as a polling location during an election, we say that it is *active*. The second is a collection of voters within the jurisdiction that must be assigned to vote at a polling location. We assume that voters are grouped using spatially delimited boundaries called *population districts*, such as voter precincts, census tracts, or census block groups. We further assume that voters within each population district must be collectively assigned to vote at the same polling location, otherwise a smaller spatial representation of voters can be used. Let I represent a set of spatially delimited population districts, and let each population district represent p_i voters.

In most jurisdictions, there is a predetermined assignment of voters to polling locations that is used during an election if possible. Let $j(i) \in J$ represent this *standard polling location* of population district $i \in I$. The standard assignment of population districts to polling locations may not always be feasible due to changing election conditions. Within the PLCP, we denote the assignment of population district $i \in I$ to polling location $j \in J$ using the binary variable $x_{ij} \in \{0, 1\}$. When $x_{ij} = 1$, population district $i \in I$ is assigned to vote at polling location $j \in J$. Otherwise, it takes a value of zero. Given the nature of consolidation decisions, we require that if the standard polling location, $j(i)$, for population district $i \in I$ is active, then i must be assigned to vote at $j(i)$ ($x_{ij(i)} = 1$). This requirement also simplifies the consolidation decisions and avoids the possibility that a voter arrives to their standard polling location and waits in line unaware that they have been reassigned to vote at another location. Under this condition, the value of the assignment variables indicate if and how polling locations are consolidated. If $x_{ij(i)} = 0$ for some population district $i \in I$, then its standard polling location $j(i)$ is not active, and it is consolidated with some polling location $j \in J$ for which $x_{ij} = 1$. In cases where multiple population districts share the same standard polling location, we allow each population district to be assigned to a unique polling location when their standard polling location is not active.

There are many constraints and criteria that election officials should follow that influence

the consolidation decisions. First, it is possible that specific polling locations, $\hat{J}$, cannot be used in the upcoming election, since the building is reserved for another use, is under construction, or is in a susceptible location (e.g., to a wildfire). The parameters and variables associated with the locations in $\hat{J}$ can be removed from the model, but we retain them for ease of discussion. Second, election conditions may require that no more than n polling locations are active. In most elections, n represents the standard number of active polling locations. However, n may be lower when election officials must reduce the cost of the in-person voting system [211] or due to legal requirements (e.g., a shortage of chief election inspectors).

The scarcity of voting resources also influences consolidation decisions. We consider two classes of voting resources that are critical to the voting process. The first are *server resources* (e.g., voting machines) of which there is a limited supply, $s_{\max}$, and these resources must be efficiently allocated to the polling locations as a result. The second are non-server *resources* (e.g., parking spaces, curbside voting space, check-in booths) that do not need to be allocated to active polling locations due to a fixed availability of the resources at each location or a sufficient global supply to distribute to the locations.

We first discuss non-server resources. Let R be a set of the different resource types. These resources do not need to be allocated to polling locations within the PLCP, but each polling location $j \in J$ has a limited capacity $\bar{a}_{jr}$ for each resource $r \in R$ within the physical space (e.g., 50 parking spaces). Let population district $i \in I$ require a_{rij} of resource type $r \in R$ if assigned to polling location $j \in J$. The values of a_{rij} and $\bar{a}_{jr}$ may change between elections and render a previously used polling location unusable. Moreover, the values of a_{rij} and $\bar{a}_{jr}$ in-part determine which consolidation schemes are feasible, since some assignments may exceed the capacity of the polling location.

We next consider the server resources. Similarly to non-server resources, we assume that each polling location $j \in J$ can utilize at most c_j server resources due to the limited physical

space of the polling location. However, the actual number of server resources available at each polling location must be determined within the PLCP. We introduce the binary variable y_{jm} to indicate whether at least $m = 1, 2, \ldots, c_j$ server resources are allocated to polling location $j \in J$. Note that the value of y_{j1} indicates whether polling location j is active ($y_{j1} = 1$) or not ($y_{j1} = 0$) during the election. The feasibility of consolidation decisions with respect to non-server resources is determined by the stochastic nature of the in-person voting process on Election Day. We assume that the server resources represent the bottleneck of the voting process at which most waiting occurs. Thus, a sufficient number of server resources must be allocated to achieve the goal set by the Presidential Commission on Election Administration that no voter waits longer than 30 minutes to vote [150].

We generalize the goal set by the Presidential Commission on Election Administration to be that no more than $1 - \alpha$ (e.g., 0.05) of the voters at each polling location wait longer than τ minutes (e.g., 30 minutes). We assume voters arrive from each population district to their assigned polling location according to a time-homogeneous Poisson process[2] and no balking occurs. Let population district $i \in I$ have arrival rate to vote of λ_i, which is determined by the number of voters expected to vote in the upcoming election. Based on the assignment of population districts to polling locations, the effective arrival rate to polling location $j \in J$ is $\lambda'_j := \sum_{i \in I} \lambda_i x_{ij}$, which follows from the merging of independent Poisson processes. Once arriving to their polling locations, voters are processed by the server-resources allocated to that location with a service time that is exponentially distributed. Under these assumptions, each polling location can be modeled as a $M/M/c$ queue[3] with an endogenous number of server resources at each polling location, c.

We employ the technique introduced in [118] to formulate a linear integer chance

[2] In reality, the arrival rate may be time-inhomogeneous. We can let the arrival rate in our model represent the peak arrival rate; however, this is a conservative assumption.

[3] In reality, a polling location is more accurately described by tandem $M/M/c$ queues with multiple server resource types (e.g. polling locations, voting machines); however, our assumption is reasonable for most contexts that would require the consolidation of polling locations.

constraint[4] that reflects our wait time goal for each polling location. By considering each polling location separately, we ensure that the resource allocation decisions are equitable. Assuming the queueing system is in steady state, the probability that the wait time w_j is greater than τ for a random voter entering the queue at polling location $j \in J$ with $m \in \{1, 2, \ldots, c_j\}$ server resources each with a service rate of μ_j is described by [31]:

$$P(w_j > \tau) = \left(\frac{m\rho_m}{m - \lambda'_j/\mu_j}\right) e^{-(m\mu_j - \lambda'_j)\tau}$$

where

$$\begin{cases} \rho_0 = \left[\sum_{k=0}^{m-1} \frac{(\lambda'_j/\mu_j)^k}{k!} + \frac{(\lambda'_j/\mu_j)^m}{(m-1)!(m-\lambda'_j/\mu_j)}\right]^{-1} \\ \rho_k = \left(\frac{(\lambda'_j/\mu_j)^k}{k!}\right)\rho_0 \qquad\qquad 1 \leq k \leq m \end{cases}$$

The wait time goal can then be restated as:

$$\left(\frac{m\rho_m}{m - \lambda'_j/\mu_j}\right) e^{-(m\mu_j - \lambda'_j)\tau} \leq 1 - \alpha$$

Root-finding methods can identify a parameter $\lambda_{\alpha m}$ such that the previous inequality holds at equality if m resources are allocated to $j \in J$. Thus, the wait time goal can be restated using the constraint $\sum_{i \in I} \lambda x_{ij} \leq \lambda_{\alpha m}$ for all $i \in I$ [118]. Since the number of resources distributed to polling location $j \in J$ is endogenous to the PLCP, the constraint can be formulated as

$$\sum_{i \in I} \lambda_i x_{ij} \leq y_{j1}\lambda_{\alpha 1} + \sum_{m=2}^{c_j} y_{jm}(\lambda_{\alpha m} - \lambda_{\alpha(m-1)}) \quad \forall\, j \in J$$

The same procedure can be used for queues with different service distributions when a reasonable approximation for the waiting time distribution can be found. For example, van Hoorn and Tijms [207] present a method to approximate the waiting time distribution for a $M/G/c$ queue.

[4] This constraint is alluded to in [118] but never formally stated.

In addition to these constraints, the collection of population districts assigned to a polling location, termed the *polling location district*, should be contiguous. Contiguity of districts ensures that a population district is not arbitrarily assigned to a polling location that is far away while there may be closer polling locations at which the population district can vote. This increases the equity of the consolidation decisions and makes them more justifiable. Contiguity is common within political districting problems [205], but the definition of contiguity does not directly extend to the PLCP. Thus, we formalize a modified definition[5] of contiguity for the context of polling location districts. First, let $N_i \subseteq I$ denote the neighboring population districts to population district $i \in I$. Neighboring population districts share a portion of their spatial boundary. Let us have a contiguity graph, G, that describes spatial relationship of population districts. A contiguity graph contains a node for each population district $i \in I$ and an edge between each neighboring population district N_i [205]. Let $i(j) \in I$ denote the population district in which polling location $j \in J$ is located. We define[6] a polling location district to be *contiguous* if for any $i \in I$ that is assigned to vote at polling location $j \in J$, there exists a path in G between i and $i(j)$ where each population district on the path (not including $i(j)$) is also assigned to j. Note that this definition does not require that the population district $i(j)$ be assigned to polling location j. This definition also allows for the possibility that two or more *polling location subdistricts* are assigned to vote at $j \in J$, and these subdistricts represent distinct connected components in G. We use this definition of contiguity, since there may be instances where some population districts have no potential polling locations within their boundary while a nearby population district may have multiple.

In order to enforce contiguity within the PLCP, we employ a modified version of the CUT formulation used to impose contiguity of political districts in [205]. Our approach relies

[5] Also called adjacency or dual graph.
[6] This is similar to a definition in [104].

on the idea of an (i, j)-separator, for which a formal definition is provided in Definition 3.1. Intuitively, an $(i, j)-$separator is a set of population districts among which at least one must be assigned to polling location j if population district i is assigned to j.

Definition 3.1 ((i,j)-Separator)**.** *Given $i \in I$ and $j \in J$, a subset of elements $C \subseteq I \setminus \{i, i(j)\}$ is an (i, j)-separator if all paths from i to $i(j)$ in the contiguity graph contain at least one node corresponding to an element in C. The (i, j)-separator is minimal if $C \setminus \{k\}$ is not an (i, j)-separator for all $k \in C$.*

Using (i, j)-separators, we can enforce contiguity. However, instantiating a constraint for each valid (i, j)-separator, C, introduces an exponential number of constraints. In Section 3.3.2, we discuss an algorithm to identify violated constraints that can be used within a lazy constraint scheme.

The dynamics of the PLCP discussed up to this point determines the feasibility of consolidation decisions. We assess feasible consolidation decisions according to the burden placed on voters in terms of the additional distance (or time) that voters must travel to their assigned polling location when compared to their standard polling location. Let d_{ij} denote the distance (or time) from population district $i \in I$ to polling location $j \in J$. Then, $\max\{0, d_{ij} - d_{ij(i)}\}$ represents the *increased distance* associated with assigning population district $i \in I$ to polling location $j \in J$. If the distance traveled by the voters in a population district decreases due to the updated assignment, then the increased distance is zero. In this way, the objective does not incentivize changing to polling location districts when no changes need to be made. The objective of the PLCP is then to identify polling location districts that minimize the value of $\sum_{i \in I} \sum_{j \in J} p_i \max\{0, d_{ij} - d_{ij(i)}\} x_{ij}$. The value of $\max\{0, d_{ij} - d_{ij(i)}\}$ can be pre-computed for each $i \in I$ and $j \in J$.

In total, these constraints and the objective function form the following integer pro-

gramming formulation of the Polling Location Consolidation Problem (PLCP):

$$\min_{x,y} \sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}x_{ij} \tag{3.1}$$

$$\text{s.t. } y_{j1} = 0 \qquad \forall j \in \hat{J} \tag{3.2}$$

$$\sum_{j\in J} y_{j1} \leq n \tag{3.3}$$

$$\sum_{j\in J} x_{ij} = 1 \qquad \forall\, i \in I \tag{3.4}$$

$$x_{ij} \leq y_{j1} \qquad \forall\, j \in J, i \in I \tag{3.5}$$

$$y_{j(i)1} \leq x_{ij(i)} \qquad \forall\, i \in I \tag{3.6}$$

$$\sum_{i\in I} a_{rij} x_{ij} \leq \bar{a}_{rj} \qquad \forall\, j \in J, r \in R \tag{3.7}$$

$$\sum_{i\in I} \lambda_i x_{ij} \leq y_{j1}\lambda_{\alpha 1} + \sum_{m=2}^{c_j} y_{jm}(\lambda_{\alpha m} - \lambda_{\alpha(m-1)}) \qquad \forall\, j \in J \tag{3.8}$$

$$\sum_{j\in J}\sum_{m=1}^{c_j} y_{jm} \leq s_{\max} \tag{3.9}$$

$$y_{jm} \leq y_{j(m-1)} \qquad \forall\, j \in J,\ m = 2, \ldots, c_j \tag{3.10}$$

$$x_{ij} \leq \sum_{k\in C} x_{kj} \qquad \forall\, i \in I, j \in J, C \tag{3.11}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, i \in I, j \in J \tag{3.12}$$

$$y_{jm} \in \{0, 1\} \qquad \forall\, j \in J,\ m = 0, 1, 2, \ldots c_j \tag{3.13}$$

The objective (3.1) of the PLCP is to minimize the population weighted increased distance between the population districts and their assigned polling locations. Constraint set (3.2) ensures that the polling locations in $\hat{J}$ are not active. Constraint (3.3) ensures no more than n polling locations are active. Constraint set (3.4) ensures that each population district is assigned a single polling location. Constraint set (3.5) requires that if a population district is assigned to a polling location, then the polling location is active. Constraint

set (3.6) requires a population district to be assigned to its standard polling location if the standard polling location is active. Constraint set (3.7) ensures that a polling location has sufficient capacity to support the population districts assigned to it with respect to (non-server) resources. Constraint set (3.8) ensures that no more than $1 - \alpha$ voters wait longer than τ minutes to vote given the number of server resources allocated to each polling location. Constraint (3.9) limits the number of server resources that can be allocated to the active polling locations. Constraint set (3.10) is a functional constraint that requires that if at least m server resources are allocated to a polling location, then at least $m - 1$ are allocated. Constraint set (3.11) ensures that the polling location districts are contiguous by requiring the assignment is valid for all (i, j)-separators. Constraint sets (3.12) and (3.13) limit the variables to be binary.

3.3.2 Enforcing Contiguity

In some districting problems, the objective function and other constraints result in naturally contiguous optimal solutions without the explicit inclusion of contiguity constraints within the model [9]. However, in general[7], the contiguity constraints (3.11) of the PLCP are required to identify a contiguous consolidation scheme. This idea is formalized in Remark 3.1 and a proof by counterexample is provided in Appendix 7.

Remark 3.1. An optimal solution to the PLCP without constraint sets (3.11) is not guaranteed to be feasible to the PLCP.

However, constraint set (3.11) defines an exponential number of constraints, and we wish to avoid instantiating all of them. A solution to the PLCP can be found by employing lazy constraints to enforce constraint set (3.11). Within this subsection, we introduce Algorithm

[7]There are specific instances in which we can guarantee that the contiguity constraints are redundant. A constraint is redundant if it can be omitted from the IP without changing the set of feasible solutions. We discuss one in Appendix 8.2.

1 based on the work of Validi et al. [205] but modified for the PLCP to identify violated (i, j)-separators given an incumbent solution, $x \in \{0,1\}^{|I|\times|J|}$ and $y \in \{0,1\}^{|J|\times c_j}$, to the PLCP without constraint set (3.11).

Algorithm 1 Lazy(G, y^*, x^*)

for $j \in J$ **do**
  if $y^*_{j1} = 1$ **then**
    $T_j := \{i \in I : x^*_{ij} = 1\} \cup \{i(j)\}$
    for every connected component G' of $G[T_j]$ that does not contain $i(j)$ **do**
      Let C_ℓ be the nodes in the connected component G'
      Let i be a random element of C_ℓ
      Let $A(C_\ell)$ denote the nodes in G neighboring those in C_ℓ that do not belong to C_ℓ
      Let C be the members of $A(C_\ell)$ found by running a breadth-first search on G starting at $i(j)$ but never adding any nodes from $A(C_\ell)$ to the queue [71]
      Add constraint $x_{ij} \leq \sum_{k\in C} x_{kj}$
    end for
  end if
end for

The procedure works as follows. For each active polling location j, identify the population districts assigned to it. Let T_j represent this collection of population districts. Add $i(j)$ to T_j. Next, identify the connected components of the contiguity graph induced by the nodes T_j. For each connected component C_ℓ that does not contain $i(j)$, we find an (i, j)-separator for at least one $i \in C_\ell$. Let $A(C_\ell)$ denote the nodes of G neighboring the nodes of C_ℓ, but which are not members of C_ℓ. We can find the (i, j)-separator, C, by running a modified breadth-first search on the contiguity graph G starting with the node $i(j)$ [71]. However, the breadth-first search is modified so that when the search finds a node that is a member of $A(C_\ell)$, it is added to C, but is not added to the queue to be further explored [71]. The resulting set C is an (i, j)-separator. For any element $i \in C_\ell$, we can add the following constraint to the integer program: $x_{ij} \leq \sum_{k\in C} x_{kj}$.

The theoretical results discussed in [205] extend to Algorithm 1. Namely, given integral x^* and y^* and a planar contiguity graph G, then Algorithm 1 runs in $\mathcal{O}(|I|^2)$ time. In most

instances, contiguity graphs within districting problems satisfy the planar assumption [205].

3.3.3 Valid Inequalities

In this section, we present three sets of valid constraints for the PLCP. Adding these constraints to the formulation may improve the computational time needed to solve the PLCP using common integer programming techniques. The first set of constraints, the resource-cover inequalities, are formalized in Theorem 3.2. These inequalities result from a direct application of the knapsack covering inequalities to constraint set (3.7) within the PLCP, so we omit a proof.

Theorem 3.2 (Resource cover-inequalities)**.** *Let $D_{rj} \subseteq I$ be a resource-cover for resource $r \in R$ and polling location $j \in J$ if*

$$\sum_{i \in D_{rj}} x_{ij} > \bar{a}_{rj}$$

Then the following constraint is valid for the PLCP.

$$\sum_{i \in D_{rj}} x_{ij} \leq |D_{rj}| - 1$$

The second set of valid inequalities, the wait time-cover inequalities, are formalized in Theorem 3.3. These inequalities are similar to knapsack covering constraints, but they are modified to address the possibility that the PLCP increases the capacity of a polling location by allocating additional server resources. These valid inequalities are novel, and a proof to Theorem 3.3 is provided in Appendix 7

Theorem 3.3 (Wait time-cover inequalities)**.** *Let $W_{jm} \subseteq I$ be a wait time-cover for polling*

location $j \in J$ and $m \in \{1, 2, \ldots, c_j\}$ if

$$\sum_{i \in W_{jm}} \lambda_i x_{ij} > \lambda_{\alpha m}$$

Then, the following constraint for $m \in \{1, 2, \ldots, c_j - 1\}$ is valid for the PLCP

$$\sum_{i \in W_{jm}} x_{ij} \leq |W_{jm}| - 1 + y_{j(m+1)}$$

Note that a W_{jm} wait time-cover is also a $W_{j(m-1)}$ wait time-cover.

The final set of valid inequalities is formalized in Theorem 3.4. We utilize a value m_j^*, which represents the minimal number of server resources needed at j to satisfy the wait time constraint when population districts are assigned to their standard polling location. These valid inequalities state that if any population district $i \in I$ is assigned to $j \in J$, then at least m_j^* resources must be allocated to j. A proof to Theorem 3.4 is provided in Appendix 7

Theorem 3.4 (Server-resource inequalities)**.** *Let $m_j^* = \text{argmin}_{m=1,\ldots,c_j}\{m : \sum_{i \in I: j(i)=j} \lambda_i \leq \lambda_{\alpha m}\}$ for each $j \in J$. Then, for each $j \in J$ and $i \in I$, the following constraint is valid for the PLCP*

$$y_{jm_j^*} \geq x_{ij}$$

3.4 Characterization of Solutions

In this section, we investigate the theoretical properties of the PLCP. The proofs to the statements made within this section are provided in Appendix 7. Many of the statements rely on the idea of a *feasible assignment* of population districts to polling locations, which we now formally define.

Definition 3.2 (Feasible Assignment)**.** *The assignment* $x \in \{0,1\}^{|I|\times|J|}$ *is a feasible assignment if there exists a* $y \in \{0,1\}^{|J|\times c_j}$ *such that* x, y *satisfies constraints* (3.2)-(3.13).

We first show in Theorem 3.5 that if assigning each population district to their standard polling location, termed the *standard assignment*, is a feasible assignment, then it is optimal to do so within the PLCP. Due to this property, the PLCP does not incentivize changing polling locations or the assignment of voters to polling locations unless necessary. Thus, the PLCP can be used to determine when polling locations should be consolidated.

Theorem 3.5. *Let the standard assignment,*

$$x_{ij} = \begin{cases} 1 & j = j(i) \\ 0 & otherwise \end{cases} \quad \forall\, i \in I, j \in J$$

be a feasible assignment. Then it is optimal to the PLCP.

However, the standard assignment is not always feasible. This is trivial for PLCP instances in which standard polling locations belong to $\hat{J}$. In Remark 3.6, we state there exists PLCP instances for which the goal that voters do not wait long to vote necessitates that polling locations be consolidated.

Remark 3.6. There are instances to the PLCP for which the feasible region only contains assignments where two or more polling location districts are consolidated due to the constraints in constraint sets (3.8)-(3.10).

When the standard assignment is not feasible, the PLCP is NP-Hard, Theorem 3.7. This is shown using a reduction of the p-median problem to the PLCP. This suggests that election officials cannot rely on rules-of-thumb, such as consolidating the nearest polling locations, to identify an optimal consolidation scheme.

Theorem 3.7. *The PLCP is NP-Hard.*

Moreover, we find that the decision problem of identifying if there exists a feasible solution to a PLCP instance is NP-Complete, Theorem 3.8. This suggests that finding a feasible consolidation scheme is challenging and motivates the use of optimization to solve the PLCP.

Theorem 3.8. *Determining if a feasible solution to the PLCP exists is NP-Complete.*

In specific instances, the PLCP may be solvable in polynomial time. To show this, we rely on the idea of a *rank* $q_i(j)$, for each polling location $j \in J$ for a specified population district $i \in I$, which is formalized in Definition 3.3. The rank of polling locations formalizes the idea of ordering the polling locations based on the distance increase associated with it for each $i \in I$. In general, this ranking is not unique, and there are an exponential number of possible valid rankings. However, the ranking is unique if for a given $i \in I$, for no more than one polling location $j \neq j(i)$ is $d_{ij} \leq d_{ij(i)}$, and for each pair $j_1, j_2 \in J$, no two distances are the same, $d_{ij_1} \neq d_{ij_2}$. These conditions may be satisfied in reasonable PLCP instances.

Definition 3.3 (Ranking of Polling Locations, $q_i(\cdot)$)**.** *A mapping $q_i : J \to \{1, 2, \ldots, |J|\}$ is a ranking of polling locations for $i \in I$ if $q_i(j(i)) = 1$ and for any $j_1, j_2 \in J$, the following conditions hold:*

1. $q_i(j_1) \neq q_i(j_2)$
2. *If $q_i(j_1) < q_i(j_2)$ then* $\max\{0, d_{ij_1} - d_{ij(i)}\} \leq \max\{0, d_{ij_2} - d_{ij(i)}\}$

Using the rank of polling locations, we provide a characterization of a PLCP instance where it is optimal to assign each population district to the polling location not in $\hat{J}$ with the lowest rank in Theorem 3.9. The result is useful when standard polling locations cannot be used and election officials wish to assign population districts to the nearest polling location. The conditions of Theorem 3.9 can be verified within polynomial time when a ranking of polling locations are given.

Theorem 3.9. *Let $q_i(j)$ be the rank of each polling location $j \in J$ for each population district $i \in I$. Let $z(i) = \operatorname{argmin}_{j \in J \setminus \hat{J}} q_i(j)$ represent lowest rank polling location not in $\hat{J}$ for $i \in I$. Then the assignment*

$$x^*_{ij} = \begin{cases} 1 & j = z(i) \\ 0 & otherwise \end{cases} \qquad \forall\, i \in I, j \in J$$

is a feasible assignment for the PCLP if and only if

1. *For each $i \in I$ such that $i(z(i)) \notin \{i\} \cup N_i$, there exists a path in $G[\{i' \in I : z(i') = z(i)\} \cup \{i(z(i))\}]$ from i to $i(z(i))$, where $G[Q]$ represents the subgraph of the contiguity graph G induced by the nodes $Q \subseteq I$.*

2. *No more than n unique polling locations provide the lowest rank across all $i \in I$, $|\cup_{i \in I} z(i)| \leq n$.*

3. *$\hat{m}_j \leq c_j$ for each $j \in J$ and $\sum_{j \in J} \hat{m}_j \leq s_{\max}$ where $\hat{m}_j = \operatorname{argmin}\{m \in \mathbb{Z}_+ : \sum_{i \in I : z(i) = j} \lambda_i \leq \lambda_{\alpha m}\}$ for each $j \in J$ and $\lambda_{\alpha 0} = 0$.*

4. *$\sum_{i \in I} a_{rij} x^*_{ij} \leq \bar{a}_{rj}$ for all $j \in J,\ r \in R$.*

Moreover, if x^* is a feasible assignment, then there exists a $y \in \{0,1\}^{|J| \times c_j}$ such that x^*, y is optimal to the PLCP.

However, as already noted, the ranking of polling locations is not always unique. In general, determining if there exists a ranking that satisfies the conditions of Theorem 3.9 is equivalent to finding a feasible solution to an instance of the PLCP.

Theorem 3.9 also informs how election officials can redesign the polling locations to avoid unnecessary consolidation while still meeting election goals. The third and fourth conditions provide the exact number of voting resources needed at each polling location to be able to feasibly assign voters to the lowest rank polling location not belonging to $\hat{J}$. When these conditions are not satisfied, election officials can source and distribute extra

voting resources or change the layout of polling locations so that the consolidation scheme discussed in Theorem 3.9, which minimizes the increased distance, can be used.

3.5 Case study

Within this section, we introduce a case study using real-world data from Richland County, South Carolina, analyze optimal consolidation decisions, and identify implications for practice. Richland County's in-person voting process can be summarized as follows [177, 179]. Voters are assigned to a single polling location at which they can vote. When voters arrive to their assigned polling location on Election Day, they check-in and receive a blank ballot. They then use an ExpressVote ballot-marking system to fill out their ballots[8]. Once completed, the ballot-marking system prints on the ballot, and the voter inserts it into a DS200 optical ballot scanner, which completes the voting process.

We construct the case study instances using publicly available data. In 2020, Richland County had 265,164 registered voters within 149 voting precincts assigned to a total of 129 polling locations. We use the voter precincts in Richland County as the set of population districts, I. We use the number of registered voters in each population district as the population of that district, p_i. We use the 129 standard polling locations [168] plus five non-standard polling locations used during the 2020 primary as the set of possible polling locations, J. We refer to the five non-standard polling locations as the "extra" polling locations. Unless otherwise noted, at most $n = 129$ polling locations can be active, and no restrictions are placed on which polling locations can be used, $\hat{J} = \emptyset$. The distance, d_{ij}, was estimated using the driving duration (minutes) from Bing Maps between the population district centroid[9] and the potential polling location.

[8]The June 9, 2020 primary was the first use of the ExpressVote systems in Richland County during a major election.

[9]The "centroid" of Ward 26 was changed, since the precinct represents Fort Jackson. An approximate population center was selected for this precinct as (34.01166, -80.940973).

We assume the expected in-person, Election Day turnout is equal to that experienced during the 2016 general election, which was 42.1% on average[10], unless otherwise noted [178]. Given an expected turnout percentage, $\mathcal{T}_i$, of each population district $i \in I$, we set the arrival rate of voters to be $\lambda_i = p_i \times \mathcal{T}_i/720$ voters per minute[11]. This represents a uniform (time-homogenous) arrival rate of voters. Since actual arrivals are likely to be time-inhomogenous, we multiple each λ_i by 1.3 to account for the effects of time-inhomogeneity.

Throughout the case study, we consider voting machines to be the server resource, since they have been shown to be the bottleneck in many voting processes [221]. Unless otherwise noted, we assume $s_{\max} = 981$, which equals the number of voting machines available during the 2020 primary election. We let the service rate of each voting machine be $\mu_j = 0.2708$ voters per minute based on empirical research [186]. The maximum number of voting machines that can fit into the footprint of each polling locations, c_j, is not recorded by election officials. We estimate each c_j as follows. We first estimate the square footage of each polling location by measuring the building dimensions using Google Maps satellite view. For elementary and middle schools, we use the size of a single gymnasium (110 ft. by 60 ft.) instead of the full square footage. For high schools, we use the size of two gymnasiums. We then estimate the number of voting machines used per square foot at each polling location based on the number of voting machines actually allocated to each polling location during the election held on June 9, 2020. We then find the maximum number of machines used per square foot across all polling locations. To set c_j, we assume that each polling location can utilize up to the number of voting machines implied by their square footage and this maximum ratio. This method assumes that for at least one polling location, the number of voting machines allocated to it on June 9, 2020 represents the true c_j associated with it and each location can use the same number of machines per square

[10]The actual turnout was approximately 61%, but approximately 19% of ballots were cast using absentee voting.

[11]Polls are open from 7 AM until 7 PM on Election Day, which is 720 minutes.

foot.

We consider two types of non-server voting resources: poll workers and parking spaces. The number of poll workers needed per voter at a polling location, $a_{\text{poll workers},ij}$, is determined by South Carolina law [2]; each polling place must have 3 poll workers for each 500 registered voters assigned to vote at the polling location. We estimate the capacity of each polling location, $\bar{a}_{\text{poll workers},j}$, using the same process as for c_j except we use the poll worker distribution from the election held on June 23, 2020, which was the only available data. We estimate the number of parking spaces available at each polling location, $\bar{a}_{\text{parking},j}$, by counting the number of parking spaces at each location on Google Maps satellite view. We assume the number of parking spots required per voter, $a_{\text{parking},ij}$, from population district i depends on the distance d_{ij} to their assigned polling location j. We estimate this relationship using the number of parking spots at each polling location and the standard assignment of each population district, assuming that no voter needs to park for longer than an hour.

Based on the recommendation of the Presidential Commission on Election Administration [150], we require no more than $\alpha = 0.05$ of the voters wait longer than $\tau = 30$ minutes to vote. This requirement defines that maximal voter arrival rate, $\lambda_{\alpha m}$, at a polling location allocated m server resources. However, the construction of $\lambda_{\alpha m}$ assumes that a polling location can be described by a $M/M/m$ queue and that the system is in steady state. These two conditions could be violated within polling locations on Election Day. Thus, we use a discrete event simulation of Election Day voting to verify the wait time experienced by voters. Within the discrete event simulation, we assume that three quarters of the poll workers assigned to a polling location will check-in voters and there are two poll workers per check-in booth. We also assume there is one optical scanner for every two precincts assigned to a polling location. We assume voters arrive according to a time-inhomogenous Poisson process using rates from [223] adjusted for a 12 hour voting period,

which reflect a morning rush of voters. We assume the check-in time (minutes) follows a $lognormal(0.4784, 0.6075)$ distribution [186], the ballot marking time (minutes) follows a $lognormal(1.092886, 0.6534)$ distribution [186], and the time (minutes) to submit the ballot to a ballot scanner follows a $triangular(0.15, 0.25, 0.35)$ distribution [166].

We divide the case study results into four subsections that each investigate a specific election condition that may influence consolidation decisions. Under these election conditions, we explore the optimal consolidation decision and identify implications for practice. The results presented throughout these subsection represent solutions obtained using the Gurobi 9.1 MIP solver using 64 bit Python 3.7.7 on an Intel® Core™ i5-7500 CPU with 16 GB of RAM. A time limit of 3600 seconds was given for each problem instance[12]. The cutting planes described in Section 3.3.2 were not implemented, since implementations did not consistently improve solution times.

We wish to note that the standard assignment of voter precincts to polling locations within Richland County does not satisfy the definition of contiguity used within the PLCP. Thus, the PLCP consolidates polling locations even with baseline parameters for the voting system resulting in an objective function value (OFV) of 1,199 minutes. Additional discussion and model extensions are provided in Appendix 8.1.

3.5.1 Reduced Election Budget

Unexpected or rising election costs may cause election officials to consolidate polling locations to reduce the cost of facilitating in-person voting [211]. In this subsection, we investigate optimal consolidation decisions when the number of polling location n must be reduced. Table 3.3 presents the OFV, number of polling locations, number of precincts reassigned, and number of registered voters reassigned in optimal consolidation decisions for various values of n. Also presented in Table 3.3 are the average (avg.) and standard

[12] As a result, the results presented in this chapter may not reflect all possible solutions to the PLCP.

deviation (std.) of the expected wait time and proportion of voters waiting at least 30 minutes within the queues of the discrete event simulation of each polling location, which verify that the consolidation decisions are successful in ensuring at most 0.05 of voters experience a wait time of more than 30 minutes. We find that optimization can identify consolidation decisions under which the OFV does not increase[13] until $n = 69$. Thus, in some jurisdictions, there may be consolidation decisions that can avoid placing a large burden on voters and can reduce the cost of facilitating the election. However, consolidation may be undesirable for other reasons, such as the added short-term burden placed on voters to search for their updated polling location [18]. For n smaller than 69, the OFV, or the total additional time voters must travel to their polling location, is relatively large and represents a large burden placed on voters by consolidation. Consolidation should be avoided in these cases if possible by reducing the cost of other components of the voting system.

3.5.2 Disrupted Server Resource Supply

During some elections, the number of server resources, $s_{\max}$, available during the election may be reduced. When server resources represent voting machines, this could be a result of a cyber-physical attack on the voting system. When server resources represent poll workers, this could be a result of a pandemic. Within this section, we study the resiliency of the voting system to a disrupted supply of voting machines. Table 3.4 summarizes the relationship between the number of voting machines ($s_{\max}$) and optimal consolidation decisions. We find that the value of $s_{\max}$ substantially alters optimal consolidation decisions, and the results provide multiple implications for practice. First, optimal allocation of the voting machines can avoid the need to consolidate polling location when $s_{\max}$ is larger than 830. Past this value, it is optimal to consolidate polling locations to more efficiently utilize

[13] Compared to a contiguous consolidation scheme with $n = 129$.

Table 3.3: Details of consolidation schemes identified by the PLCP for various n while using turnout data from the 2016 general election. The wait time and proportion of voters waiting at least 30 minutes ("Prop. Wait 30 min") is estimated using 50 replications of the discrete event simulation.

	PLCP Solutions				Discrete Event Simulation	
n	Total Added Transit Time (min.), OFV	Num. Polling Locations	Num. Precincts Reassigned	Num. Voters Reassigned	Avg. (Std.) Wait time in min.	Avg. (Std.) Prop. Wait 30 min.
129	1,199.1	126	5	13,190	5.2 (0.236)	0.027 (0.005)
119	1,199.1	119	12	20,035	4.9 (0.301)	0.026 (0.006)
109	1,199.1	109	23	33,643	5.0 (0.278)	0.025 (0.007)
99	1,199.1	99	34	51,763	5.3 (0.264)	0.028 (0.007)
89	1,199.1	89	46	74,043	4.7 (0.243)	0.025 (0.005)
79	1,199.1	79	61	101,659	5.0 (0.266)	0.027 (0.007)
69	3,973.6	69	76	131,427	5.3 (0.269)	0.024 (0.006)
59	20,652.2	59	87	145,612	5.6 (0.343)	0.021 (0.007)
49	49,404.7	49	97	164,412	5.5 (0.328)	0.021 (0.008)
39	93,096.8	39	108	186,136	4.6 (0.314)	0.016 (0.008)
29	164,431.3	29	119	208,632	6.1 (0.380)	0.022 (0.010)
19	306,795.7	19	130	228,070	6.4 (0.316)	0.020 (0.008)

the voting machines. When the number of server resources available is between 830 and 790, optimal consolidation decisions do not increase the time that voter spend traveling (OFV) to their polling location[13]. This suggests that consolidating a moderate number of polling locations when a disruption to the number of server resources available occurs may allow election officials to meet election goals without substantially increasing the burden placed on voters in terms of the time they must spend traveling to their polling location. For lower values of $s_{\max}$, optimal consolidation decisions result in voter assignments that increase the OFV, indicating that some voters must spend more time traveling to their polling locations. For these values of $s_{\max}$, there is a trade-off between minimizing the increased distance that voters must travel and ensuring voters can vote quickly once arriving to their polling location. This idea is explored further later within this section.

These results imply the existence of two $s_{\max}$ thresholds below one it is optimal to

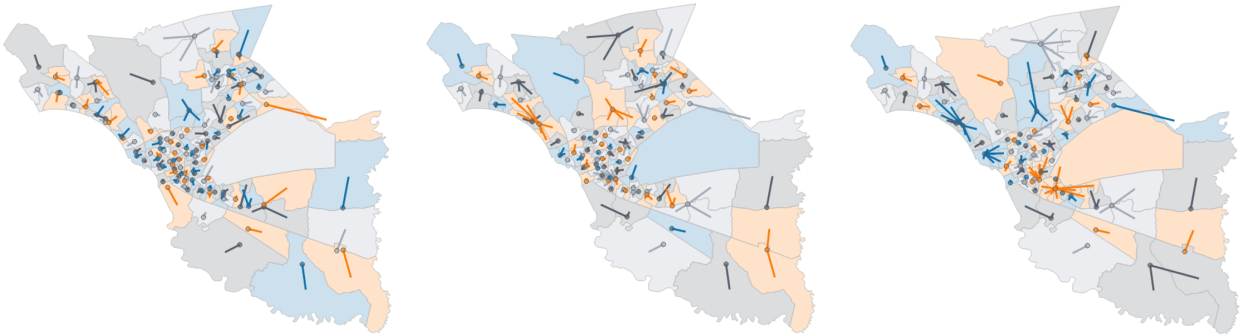
consolidate polling locations and below the other voters must travel further to their polling location. Knowing these thresholds is valuable. When known, election officials can implement mitigations to prevent situations in which the number of server resources available is below either of these thresholds. For example, election officials may consider diversifying their collection of voting machines so that not all machines are susceptible to the same vulnerabilities, although this has its own complications such as having different processes to set-up and manage different machines.

Table 3.4: Details of consolidation schemes identified by the PLCP for various $s_{\max}$ while using turnout data from the 2016 general election. The wait time and proportion of voters waiting at least 30 minutes ("Prop. Wait 30 min") is estimated using 50 replications of the discrete event simulation.

	PLCP Solutions				Discrete Event Simulation	
$s_{\max}$	Total Added Transit Time (min.), OFV	Num. Polling Locations	Num. Precincts Reassigned	Num. Voters Reassigned	Avg. (Std.) Wait time in min.	Avg. (Std.) Prop. Wait 30 min.
981	1,199.1	126	5	13,190	5.3 (0.298)	0.030 (0.006)
830	1,199.1	126	5	13,190	6.8 (0.339)	0.041 (0.009)
825	1,199.1	122	9	17,500	6.8 (0.292)	0.041 (0.007)
820	1,199.1	117	14	23,473	6.9 (0.374)	0.042 (0.010)
815	1,199.1	112	20	29,793	7.0 (0.310)	0.040 (0.008)
810	1,199.1	107	25	38,073	7.1 (0.339)	0.042 (0.008)
805	1,199.1	102	32	48,771	7.4 (0.327)	0.047 (0.009)
800	1,199.1	97	37	60,326	7.6 (0.400)	0.048 (0.009)
795	1,199.1	91	47	76,798	7.9 (0.434)	0.051 (0.012)
790	1,342.9	83	57	93,784	8.0 (0.404)	0.050 (0.011)
785	3,080.3	78	64	109,087	8.3 (0.457)	0.053 (0.012)
780	8,640.4	73	71	120,375	8.6 (0.379)	0.058 (0.012)

We explore the optimal consolidation decisions for $s_{\max} = 981,\ 795,$ and 780 in more detail in Figure 3.1. These figures can be interpreted as follows. The outer boundary represents Richland County, and each smaller polygon represents a voter precinct within Richland County. Each active polling location is represented by a small circle, and the optimal assignment of each voter precinct to a polling location is indicated by a line. The

voter precincts and polling locations of a polling location district are colored so that no two bordering polling location districts share the same color. Demonstrated by these figures, we find two patterns within optimal consolidation decisions. First, many polling locations and the voters they serve remain unchanged irregardless of $s_{\max}$. Second, there are a small number of polling locations at which a large number of population districts are assigned to vote. These polling locations are generally larger, have the capacity to serve more voters, and lie near major highways, so the additional time that voters must spend to travel to those locations is small. Many of these locations are the extra polling locations that were used during the 2020 primary election [160].



(a) $s_{\max} = 981$ (b) $s_{\max} = 795$ (c) $s_{\max} = 780$

Figure 3.1: Consolidation schemes identified by the PLCP for various $s_{\max}$ while using turnout data from the 2016 general election.

These extra polling locations are valuable to making consolidation decisions that minimize the burden placed on voters. To demonstrate the importance of these extra locations, we resolve the PLCP, but require that the extra polling locations are not active. Figure 3.2 presents the OFV of optimal consolidation schemes when the extra polling locations can be used (—△—) compared to when they cannot (—□—). We find that removing the ability to use the extra polling locations substantially increases the OFV when $s_{\max}$ is relatively small. As a result, the voting system is more susceptible to a reduction in the available number of server resources, $s_{\max}$. Thus, identifying large and strategically located back-up buildings

to use as polling locations when consolidation is necessary is an important step of election planning.

Despite a disruption to the supply of server resources, some jurisdictions may elect to retain as many standard polling locations as possible and then allocate the server resources to the locations to best minimize voter wait time. This strategy was adopted by many jurisdictions during the COVID-19 pandemic. We investigate this strategy using the PLCP by changing the wait time threshold τ used within the chance constraint (3.8). The results suggest that avoiding consolidation may not always be the best practice. Figure 3.3 plots the OFV of optimal consolidation schemes against τ for multiple $s_{\max}$. In general, we find that the OFV is larger for smaller τ and smaller $s_{\max}$, since polling locations are consolidated to better utilize the server resources. However, for values of $s_{\max}$ at or greater than 785, the increase in the OFV is near zero between $\tau = 30$ and $\tau = 60$. In these cases, consolidation can ensure voters wait less time to vote without substantially increasing the time spent traveling to their polling location. This suggests that consolidation may be a better practice than non-consolidation, since it may reduce the "cost" of voting for voters. This result is particularly important when the burden of waiting at the polling locations is much larger than the burden of traveling to the polling location, which may occur when there is poor weather (if waiting occurs outdoors) or when there is pandemic (if waiting occurs inside). The results presented in Figure 3.3 also suggest that there exists some threshold for $s_{\max}$ (e.g. 780), at or below which the decision to consolidate is non-trivial. Below this threshold, voter wait time cannot be reduced using consolidation without substantially increasing the OFV of optimal consolidation decisions. Additional empirical research is needed to understand how voters respond to each "cost" of voting to determine the policy that maximizes voter participation.

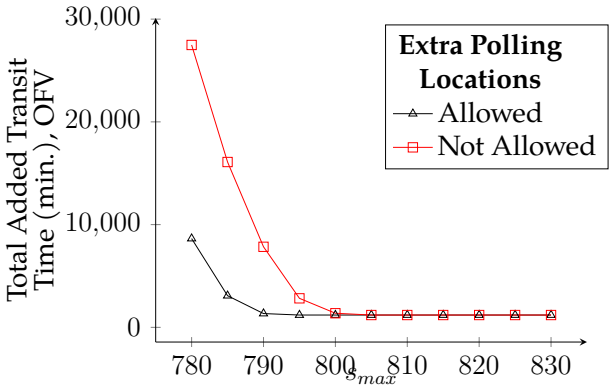


Figure 3.2: OFV of consolidation schemes for various $s_{\max}$ when $\hat{J} = \emptyset$ (—△—) and when $\hat{J}$ is comprised of the extra polling locations (—□—) when using turnout data from the 2016 general election.

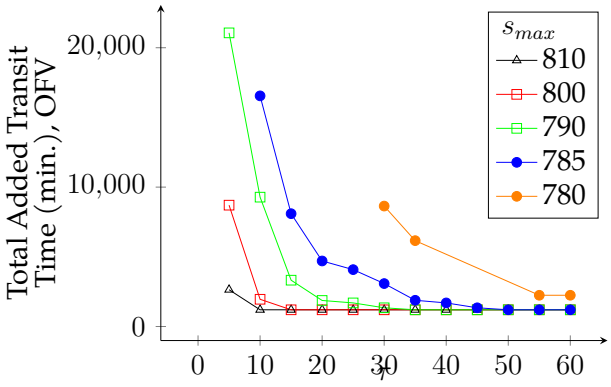


Figure 3.3: OFV of optimal consolidation schemes for various $s_{\max}$ and τ for the 2016 general election.

3.5.3 Expected Voter Turnout

Voter turnout is typically lower during a primary election than a general election, which often lowers the utilization of election resources. We investigate the optimal consolidation decisions during a primary election when expecting the in-person, Election Day turnout experienced during the 2020 primary election, which was 25.1% on average [180]. We find most practical implications are the same as those discussed in Sections 3.5.1 and 3.5.2, but we discuss one important result.

When compared to optimal consolidation decisions made when expecting a relatively high voter turnout presented in Tables 3.3 and 3.4, the consolidation decisions made for a primary election differ substantially with respects to when it is optimal to consolidate polling locations and which locations are consolidated to meet election goals. Table 3.5 presents a summary of optimal consolidation decisions for various values of n, and Table 3.6 presents a summary of optimal consolidation decisions for various values of $s_{\max}$ when the in-person, Election Day turnout experienced during the 2020 primary is expected. The results presented in Table 3.6 suggest that consolidation is not necessary during a

primary until a much larger disruption to the supply of server resources occurs ($s_{\max} = 535$ v. $s_{\max} = 825$). This is because fewer server resources are needed to ensure voters do not wait long to vote. Moreover, the results in Tables 3.5 and 3.6 demonstrate that once consolidation does occur, optimal consolidation decisions result in a lower OFV than comparable solutions in Tables 3.3 and 3.4. For example, when $n = 29$, the OFV of optimal consolidation decisions is 139,688 during a primary and 164,431 during a general election (a 17.7% increase). This is because some polling locations to which voters would be reassigned during a primary election cannot support the larger number of voters that would be seen during a general election. In this way, the burden of consolidation is higher during elections in which voter turnout is larger. As a result, contingency consolidation plans should be developed for as many election conditions as possible in order to reduce the frequency of polling location consolidations and to appropriately consolidate polling locations given the election conditions at hand.

3.5.4 A Disaster Scenario

During a disaster (e.g., hurricane) or when a disaster is imminent, election officials may be forced to quickly consolidate polling locations to reduce the complexity of managing the election and to ensure the safety of the voters. In practice, election officials may use a certain rule-of-thumb to select which polling locations are used. One rule-of-thumb is to use only schools as polling locations, since they are generally well dispersed and are large in size. We briefly assess the practice.

In Richland County, schools comprise 69 of the possible polling locations. Figure 3.4 illustrates the optimal consolidation schemes identified by the PLCP when (a) only schools can be used as polling locations or (b) at most $n = 69$ polling locations can be located, but no other restrictions are placed on which polling locations can be used. We find that the polling locations selected are substantially different. Moreover, under the school-only

Table 3.5: Details of consolidation schemes identified by the PLCP for various n while using turnout data from the 2020 primary election. The expected wait time and proportion of voters waiting at least 30 minutes ("Prop. Wait 30 min") is estimated using 50 replications of the discrete event simulation.

	PLCP Solutions				Discrete Event Simulation	
n	Total Added Transit Time (min.), OFV	Num. Polling Locations	Num. Precincts Reassigned	Num. Voters Reassigned	Avg. (Std.) Wait time in min.	Avg. (Std.) Prop. Wait 30 min.
129	1,199.1	126	5	13,190	3.3 (0.295)	0.016 (0.007)
119	1,199.1	119	12	20,035	3.1 (0.238)	0.013 (0.004)
109	1,199.1	109	23	33,643	3.5 (0.300)	0.014 (0.005)
99	1,199.1	99	34	51,137	3.4 (0.306)	0.014 (0.006)
89	1,199.1	89	46	72,516	4.2 (0.348)	0.019 (0.009)
79	1,199.1	79	60	98,760	3.0 (0.299)	0.013 (0.006)
69	2,665.1	69	77	131,619	4.0 (0.322)	0.022 (0.009)
59	16,123.1	59	87	143,663	2.9 (0.264)	0.009 (0.006)
49	42,372.6	49	98	164,279	2.0 (0.172)	0.009 (0.005)
39	80,932.3	39	109	186,047	2.4 (0.251)	0.006 (0.005)
29	139,687.8	29	120	208,188	3.8 (0.356)	0.014 (0.009)
19	246,476.4	19	131	227,865	2.3 (0.261)	0.010 (0.009)
9	677,602.1	9	142	253,573	3.2 (0.329)	0.016 (0.015)

policy, the OFV is substantially larger (217,079) when compared to the optimal policy (3,974)[14]. This suggests that rules-of-thumb may lead to very poor consolidation schemes and supports the use of the optimization models when making consolidation decisions during a disaster.

3.6 Discussion

In this chapter, we introduce a structured and transparent approach to help election officials make challenging decisions regarding polling location consolidation. We do so by formalizing the constraints and criteria that election officials should consider as the polling location

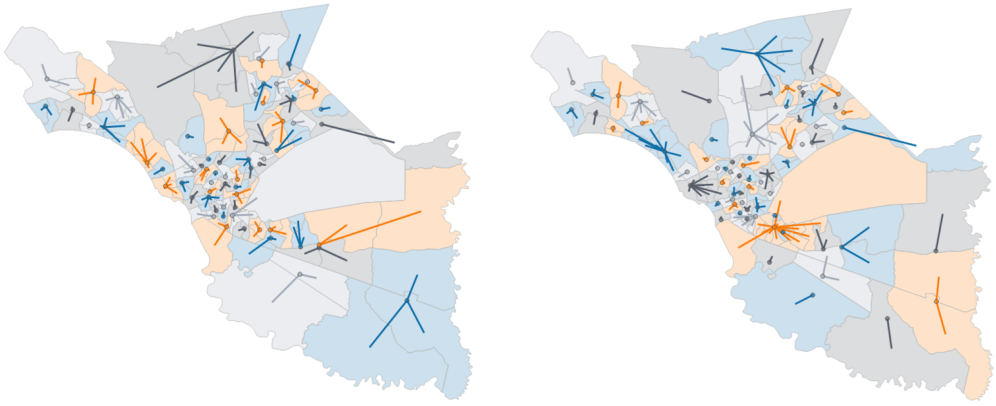
[14]Even when the extra polling locations cannot be used, the consolidation scheme identified by the PLCP is substantially better than the school-only policy (OFV of 36,996 v. 217,079, respectively).

Table 3.6: Details of consolidation schemes identified by the PLCP for various $s_{\max}$ while using turnout data from the 2020 primary election. The wait time and proportion of voters waiting at least 30 minutes ("Prop. Wait 30 min") is estimated using 50 replications of the discrete event simulation.

	PLCP Solutions				Discrete Event Simulation	
$s_{\max}$	Total Added Transit Time (min.), OFV	Num. Polling Locations	Num. Precincts Reassigned	Num. Voters Reassigned	Avg. (Std.) Wait time in min.	Avg. (Std.) Prop. Wait 30 min.
981	1,199.1	126	5	13,190	3.4 (0.266)	0.015 (0.006)
540	1,199.1	126	5	13,190	4.4 (0.283)	0.021 (0.007)
535	1,199.1	124	7	15,421	4.7 (0.329)	0.024 (0.008)
530	1,199.1	120	11	19,730	4.9 (0.334)	0.025 (0.007)
525	1,199.1	115	16	26,614	5.1 (0.337)	0.028 (0.008)
520	1,199.1	110	22	33,059	5.2 (0.373)	0.028 (0.009)
515	1,199.1	105	28	42,346	5.4 (0.415)	0.032 (0.008)
510	1,199.1	101	32	50,361	5.6 (0.325)	0.034 (0.009)
505	1,199.1	96	38	61,633	6.0 (0.434)	0.036 (0.010)
500	1,199.1	90	45	74,134	6.1 (0.455)	0.035 (0.011)
495	1,199.1	85	56	92,463	6.6 (0.377)	0.043 (0.010)
490	1,517.4	77	65	108,066	6.8 (0.373)	0.041 (0.012)
485	3,996.9	71	75	127,005	7.2 (0.453)	0.043 (0.013)
480	10,280.0	67	79	132,794	7.4 (0.556)	0.048 (0.017)
475	19,957.3	63	83	140,088	7.8 (0.537)	0.052 (0.016)
470	33,584.4	55	92	153,186	8.0 (0.496)	0.051 (0.015)
465	51,078.0	50	97	161,935	8.4 (0.528)	0.057 (0.019)

consolidation problem (PLCP). We then introduce an integer programming formulation of the PLCP to optimally determine when and how to consolidate polling locations. We study the theoretical properties of the PLCP and show that it is challenging to find feasible consolidation decisions, let alone optimal ones. This highlights the complexity of decisions currently being made by election officials and supports the use of optimization for this problem. The proposed integer program can help election official develop pre-approved contingency plans that obtain the proper legal approvals and allow election officials to be responsive to changing election conditions.

Using a real-world case study of Richland County, South Carolina, we highlight multiple



(a) Schools only. (b) Any polling location.

Figure 3.4: Comparing polling location districts during a hurricane when at most $n = 69$ polling locations can be located.

implications for practice. First, the case study demonstrates that optimal allocation of critical election resources can avoid the need for consolidation in some cases. When consolidation is necessary, optimization can often identify a small, strategic set of polling locations to consolidate in a way that minimizes additional distance that voters must travel to their polling location. Second, we demonstrate that retaining all standard polling locations and accepting a longer wait time at each may be a worse policy than consolidation in some situations. We highlight instances in which consolidation can ensure voters do not wait long to vote without substantially increasing the additional distance voters must travel to their polling location. Thus, consolidation may ultimately result in higher voter participation. However, the possibility of identifying consolidation decisions that have a minimal impact on the election system relies on the voting infrastructure. We find that large, strategically located buildings that can be used as a polling location when consolidation is necessary substantially reduces the burden placed on voters by consolidation. As such, a critical step of election infrastructure planning is identifying these locations. Most importantly, our research demonstrates the value of contingency planning using optimization. Rules-of-thumb often overlook many of the election constraints and criteria that election officials

should follow and may ultimately lead to voter disenfranchisement.

The case study also suggests that the Presidential Commission on Election Administration's goal that voters should not have to wait longer than 30 minutes to vote may be overly simplistic for cases in which polling locations must be consolidated. The guidance ignores the possibility that consolidation must occur to achieve this wait time goal. We demonstrate that ensuring voters experience a short wait once arriving to their polling location can substantially increase the time voters spend traveling to their polling location in some cases. However, determining which "cost" has a larger influence on voter participation requires further empirical research.

This work represents the first research into the polling location consolidation problem, and much work is left for the future. Within this chapter, we focus on minimizing the disruption to the entire voting system. However, the resulting decisions may negatively impact voters in certain areas or of certain socioeconomic backgrounds more than others. Additional research is needed to understand how consolidation decisions can be made more equitably. We also assume throughout this chapter that all necessary information to solve the PLCP is available at the time of planning and the consolidation decisions are made once during an election period. However, election officials may consolidate polling locations sequentially in order to "wait and see" if it is actually necessary to consolidate the polling locations considering changing election conditions. This adds additional complexity to the PLCP that may change optimal consolidation decisions. We also assume that each polling location can be modeled using a $M/M/c$ queue, and the case study suggests that this approach can produce reasonable results. However, a polling location is actually a tandem queueing system for which the allocation of multiple servers types (i.e., poll workers and voting machines) impact the wait time experienced by voters. Each of these resources consume a specific amount of the physical footprint, and there is a trade-off between adding one of each type of the server resources at a polling location. Further theoretical research

is needed to understand how these queueing dynamics can be incorporated within the optimization model. Lastly, there are many instance of the PLCP that the Gurobi state-of-the-art solver is not able to solve within 3600 seconds. Additional research is needed to understand how to reduce the solution time for challenging instances.

Chapter 4

An optimization model for locating ballot drop boxes

4.1 Introduction

During the 2020 General election, a record 46% of voters cast a ballot by mail or absentee in-person [131]. Approximately 41% of these voters cast a ballot using a drop box [152], which are temporary or permanent fixtures similar to United States Postal Service (USPS) postboxes. Many states increased the number of drop boxes during 2020 in response to increased use of the vote-by-mail system and to help mitigate health risks associated with in-person voting [44]. In total, forty states and Washington, D.C. allowed some use of ballot drop boxes [94]. However, the increase in drop box use is likely not a one time event. The use of non-traditional voting methods within the United States has steadily grown since 1996 [165]. A recent survey of Wisconsin election clerks found that approximately 78% of election clerks would like some use of ballot drop boxes in future elections, and this percentage is higher among clerks from jurisdictions with a large voting age population [23]. Many states have since introduced legislation to expand the number of drop boxes

available to voters[1] [210].

Reasons for casting a ballot using a drop box include the perceived security they offer, anticipated mail delays, and a lack of voter confidence in the USPS [153]. For many individuals, drop boxes are also in close proximity of their home, work, or daily routine [187]. Arguably, the primary benefit of drop boxes is the increased accessibility they offer to the voting infrastructure compared to in-person voting. Studies suggest that adding drop boxes to a voting system can increase voter turnout [40, 122]. McGuire et al. [122] found that a decrease in one mile to the nearest drop box increases the probability of voting by 0.64 percent. This finding aligns with the hypothesis of election participation first offered by Downs [56]. According to this hypothesis, potential voters decide whether to vote by comparing the cost (e.g., time) of voting and the potential benefits from voting. It was later argued that voting cost is the significant driver of voter turnout [170, 88]. We posit that the election infrastructure plays a large role in determining the cost to vote [26, 122, 40]. Thus, if we can improve the accessibility of ballot drop boxes to voters by appropriately designing the drop box infrastructure, then we can increase voter participation, particularly among groups who previously had a high cost to vote and low turnout.

Although drop boxes can increase voter participation, there are many challenges associated with identifying drop box locations and managing the drop box voting system. First, drop boxes can pose a large financial cost. Drop boxes can cost $6,000 [99], and designated video survillance cameras that increase drop box security can cost up to $4,000 [164]. Second, with an increased number of drop boxes, substantial time and resources must be devoted to collecting ballots. During the election period, it is recommended that bipartisan teams regularly collect ballots [99]. If drop boxes are not strategically placed or if there are a large number of drop boxes, this route may be costly and leave less time to devote to other election tasks. Third, there are security risks associated with ballot drop

[1]There are challenges to some proposals and even calls to restrict the use of these resources [210].

boxes that must be addressed [163], although drop boxes are considered reliable [184, 163]. If the drop box specific security risks are mitigated appropriately, adding drop boxes to a voting system makes an adversarial attack on the electoral process more challenging. This improves the overall security of the voting system, since the system becomes more distributed [163]. In addition to the previously mentioned challenges, elections are administered by state and local governments, and each may have different voting processes. While the vote-by-mail process is typically similar across difference jurisdictions, each jurisdiction may have unique challenges or preferences that necessitates a detailed analysis of potential drop box system design.

In light of these complexities, existing guidelines for selecting drop box locations are often insufficient to support election administrators. In 2020, the Cybersecurity and Infrastructure Security Agency [99] recommended that a drop box be placed at the primary municipal building, there be a drop box for every 15,000–20,000 registered voters, and more drop boxes should be added where there may be communities with historically low absentee ballot return rates. However, these guidelines are not prescriptive enough to support administrators in identifying an appropriate portfolio of drop box locations. To our knowledge, the only analytical approach to selecting drop box locations uses a Geographic Information System (GIS) to determine the locations that served the most voters, allowing for a maximum drive time of 10 minutes [83]. This approach overlooks many of the trade-offs within the voting system and ignores socioeconomic differences between voters that may make voting more challenging for some individuals.

Without adequate decision support tools, election administrators may ultimately select drop box locations that perform poorly across multiple criteria by which voting systems are measured. In this chapter, we propose an integer program (IP) to support election administrators in determining how ballot drop boxes should be used in their voting systems

when allowed by law[2] We formalize the IP as the drop box location problem (DBLP). To our knowledge, the DBLP is the first mathematical model of the ballot drop box system to support election planning. The DBLP seeks to minimize the capital and operational cost of the drop box system, ensure equity of access to the voting system, and mitigate risks associated with the drop box system. Loosely, we let *access* refer to the proximity of the voting infrastructure (e.g., polling places, drop boxes) to voters and the ease with which voters can cast a ballot. Expanding access through the use of drop boxes is an important aspect of the DBLP, since voter turnout is highly correlated with the distance needed to travel to cast a ballot [26]. We measure access to the drop box voting system using conventional covering sets. In addition, we propose a function based on concepts from discrete choice theory to measure the level of access a voter has to the multiple voting pathways offered by the voting system.

The remainder of the chapter is structured as follows. In Section 4.2, we review the management science literature related to elections. In Section 4.3, we discuss measures by which the ballot drop box system can be assessed. We then formalize the drop box location problem (DBLP) and introduce an IP formulation of the DBLP. In Section 4.4, we discuss solution methods for the DBLP and introduce a heuristic to quickly generate a collection of feasible solutions for election officials to select from a posteriori. In Section 4.5, we introduce a case study of Milwaukee, WI using real-world data. Using this case study, we demonstrate the value of our integer programming approach compared to rules-of-thumb that may otherwise be used. We find that the DBLP outperforms the rules-of-thumb with respect to nearly all criteria considered. We then investigate the trade-off between cost, access, and risk within potential drop box system designs. We find that the trade-off is non-trivial, and the optimization-based approach provides value. We conclude with a brief discussion in Section 4.6.

[2]The ability to use or not use drop boxes and in what capacity is typically set by state law.

4.2 Literature Review

Much of the management science literature aimed at supporting election planning focuses primarily on in-person voting processes. Some research focuses on identifying and describing the in-person voting process including quantifying the arrival rate of in-person voters, the attrition rate of polling place queues, the check-in service rate, the time to vote, and poll worker characteristics [181, 186]. This research also studied how voting requirements (e.g., the introduction of voting identification requirements) impacts voting times [186]. Queueing theory has been widely used to analyze lines at polling locations and identify mitigating practices to avoid long lines [190, 166]. Since voting machines have been recognized as a bottleneck in the in-person voting process [221], a stream of papers has focused on the allocation of voting machines to polling locations [6, 214, 65].

Other research has focused on risks of voting systems rather than operational design. The Election Assistance Commission (EAC) [63] analyzed threats to voting processes in the U.S. for seven voting technology types. Scala et al. [163] identified security threats for mail-in voting processes and offered a relative score for each to identify the most important threats to address. They identify three drop-box related threats. First, a misallocation of drop boxes can suppress voter turnout. Second, a drop box can be damaged or destroyed. Third, ballots within a drop box can be stolen or manipulated. They find the likelihood of drop box risks to be relatively low compared to other risks [163]. Fitzsimmons and Lev [72] study geographic-based risks by introducing a control problem to study how voter turnout can be manipulated through the strategic selection of polling locations. A few papers attempt to detect disruptions or security incidents following an election [92, 6].

There are no known papers intended to support election administrators in planning and managing the vote-by-mail system. Our proposed integer program addresses the risks of the drop box system [163] and employs concepts from the facility location literature.

Facility location problems are defined by a set of demands (e.g., voters) and a set of facilities (e.g., drop boxes) that can serve the needs of the demands. Arguably the most widely used facility location model is the maximal covering location problem (MCLP) [36]. In the MCLP, a demand is "covered" by, or can be served by, a predetermined set of locations called the *covering set*. Facility locations are selected to maximize the number of demands covered by at least one facility. The location set covering problem (LSCP) instead requires that all demands are covered and the cost of the selected facility locations is minimized [197].

The IP introduced within this chapter extends the covering tour problem (CTP) [78], which is a variant of the LSCP, by considering additional constraints and objective function terms. These changes allow us to accurately model the drop box voting system. A CTP instance is defined by an undirected weighted graph with two mutually exclusive and exhaustive set of nodes, the *tour* nodes and *coverage nodes*. The objective of the CTP is to find a Hamiltonian tour of minimal length over a subset of the tour nodes such that each coverage node is covered by at least one node visited by the tour. The CTP is NP-Hard since the traveling salesman problem (TSP) can be reduced to it [78]. Several solution methods, including exact [78, 13] and heuristic [137, 209], have been proposed for the CTP. This chapter represents the first known application of a CTP variation to voting systems.

4.3 Problem Definition

Election administrators face many questions regarding the use of ballot drop boxes including whether drop boxes should be added to their local voting system and how drop boxes may affect voting performance measures. If election administrators decide to add drop boxes, they must decide how many drop boxes to add and where they should be located. The DBLP introduced in this section identifies the optimal placement of drop boxes once

election administrators decide to add drop boxes to the voting system. However, election administrators can use the model during the election planning process to assess the cost, access, and risk of a potential drop box system. This can inform their decision of whether or not to add any drop boxes to the voting system.

The decisions surrounding the use of drop boxes are complex due to the number of potential locations for drop boxes, concerns about equity within the voting process, and the multiple criteria by which voting systems are measured. The most widely reported election performance metrics are the number of individuals registered to vote and the fraction of eligible voters that cast a ballot, known as *voter turnout* [132]. In most states, there are multiple pathways by which voters can cast a ballot, and the accessibility of each pathway can influence voter turnout. Figure 4.1 describes the two main pathways, which are typically divided into 'in-person' or 'absentee'. With in-person voting, a voter obtains and casts a ballot at their assigned polling location, typically on Election Day. With absentee voting[3], a voter requests a ballot be sent to them and the completed ballot is then returned either through the mail or using a drop box. In some states, voters must provide a reason to vote absentee, while in others there is "no-excuse" absentee voting.

In addition to voter-based election metrics, the cost and security of the voting system is a key concern. The cost of an election is comprised of both infrastructure-based costs (e.g., polling locations) and resource based costs (e.g., staff). The security of a voting system is not typically measured or reported to the public, despite being a major concern of officials and the public.

In this chapter, we are concerned with a sub-pathway of the vote-by-mail process where the voter submits a ballot using a drop box. In this pathway, a voter first requests and receives a ballot through the mail. They then decide to submit a ballot using a drop box

[3]Some states, such as Washington, use the "absentee" voting process as their primary voting method. Thus, we use "absentee" loosely in this chapter, and sometimes refer to it as the vote-by-mail process.

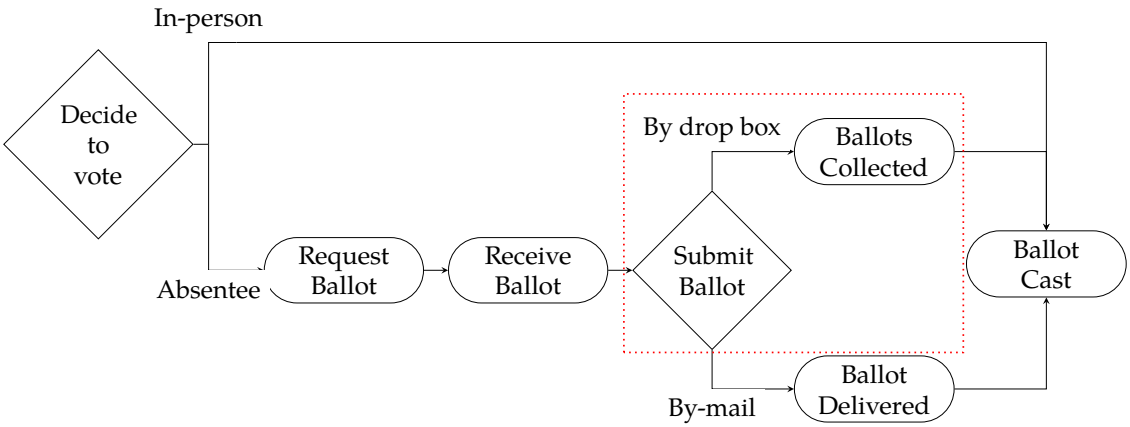


Figure 4.1: Typical pathways to cast a ballot, divided into in-person and absentee, and the component corresponding to the use of ballot drop boxes ( ).

rather than through the mail (or not returning it at all). This decision is influenced by the proximity of a drop box to the voter and the distrust the voter has in the USPS [122]. A team of poll workers then collects ballots from the drop boxes, and the ballots are processed at an official election building. In this chapter, we focus on the system related to the steps outlined in red ( ), since they are the steps that are unique to the drop box system and are influenced by the locations of the drop boxes.

4.3.1 Assessing Drop Box Infrastructure

There are two metrics typically used to assess the vote-by-mail system: the proportion of requested ballots that are returned and the number of ballots rejected [132]. The use of ballot drop boxes can lower the rejection rate of mail ballots by reducing the time it takes a ballot to return to election officials. As a result, a voter can be notified of an incorrectly marked ballot more quickly to allow the voter to resubmit their ballot before the election deadline. This is a benefit that we do not explicitly consider in our model. We also posit based on empirical research that a well-designed drop box system can lead to a higher proportion of returned mail ballots and a higher voter turnout by improving the accessibility

of the voting infrastructure [56, 122].

We elaborate on how access to the voting system is measured. We employ the concept of *coverage* to measure the access voters have to the drop box system. Under the concept of coverage, a voter covered by a selected drop box location is assumed to have access to the drop box voting system. The locations that provide a voter coverage are called its *covering set*. Covering sets are flexible and can be defined to account for different modes of transportation, vehicle ownership, and other socioeconomic factors. However, drop boxes are a subcomponent of a larger voting system, and coverage overlooks the access provided by non-drop box voting pathways. In reality, some individuals may have better access to in-person voting than others, and adding drop boxes near them may not substantially benefit them. This necessitates a second measure of access that distinguishes access to the complete voting infrastructure from coverage by the drop box system.

We introduce an access function based on the multinomial/conditional logit model from discrete choice theory [7] to capture this dynamic. The application of discrete choice theory to questions within political science is most commonly used to explain or predict choices within a multi-candidate (or party) election [81]. Discrete choice models have also been used to predict how individuals interact with infrastructure in different application domains. One of the earliest cases of this was the application of a conditional logit model to predict the use of the Bay Area Rapid Transit prior to its construction [198]. To the best of our knowledge, our chapter represents the first use of a function based on discrete choice theory to model access within an optimization model.

The function we introduce makes use of some parameters. Let $v_w^1 > 0$ be a measure of accessibility[4] to the non-drop box voting system (e.g., in-person polling locations) for voters w. This can be determined, for example, by the distance to the nearest polling location. Let

[4]In its exact form, $v_w^1 = e^{U_w^1}$ where U_w^1 represents the utility of voting using the non-drop box voting system.

$a_{nw} > 0$ be a measure of the access[5] that a drop box at location n would provide to w. This can be determined in part by the proximity of the location to the voters' places of residence and work and by the various transportation modes available between the voters and the drop box location. Based on empirical studies, the value of a_{nw} should be increasing with decreasing distance [122]. Finally, let $v_w^0 > 0$ be the propensity of w not to vote[6]. This could be informed by the historical non-voting rate (complement of turnout) or using surveys. Using these parameters, we introduce the following *access function* to measure the access a group of individuals w has to all voting pathways where N^* represents the set of selected drop box locations:

$$A_w(N^*) := \frac{v_w^1 + \sum_{j \in N^*} a_{jw}}{v_w^0 + v_w^1 + \sum_{j \in N^*} a_{jw}}$$

The access function takes values between zero and one. A value closer to one means that the voting system, including the new ballot drop boxes, is more accessible to individuals w, whereas value closer to zero means that the voting system is relatively inaccessible to individuals w. In this way, a higher access function value suggests higher turnout for w.

The access function can still be used when a strict interpretation is not reasonable or is not feasible due to data availability, since the benefit of the access function is a result of its structure. First, the access function models access as a non-binary measure. Second, adding any drop box to the voting system increases the value of the access function but to varying degrees based on the locations of the voter and the drop box. Third, each voter has some heterogeneous level of access to non-drop box voting methods captured by v_w^1, and this access is treated as a constant within the scope of the decision to location drop boxes. Each voter also has a heterogeneous access function value when no drop boxes are added to the voting system, $A_w(\emptyset) = \frac{v_w^1}{v_w^0 + v_w^1}$, which is reflective of heterogeneous turnout rates.

[5] In its exact form, $a_{nw} = e^{U_{wn}}$ where U_{wn} represents the utility of voting by using drop box n.

[6] In its exact form, $v_w^0 = e^{U_w^0}$ where U_w^0 represents the utility of not voting.

Fourth, the benefit of adding a drop box near a voter is marginally decreasing as the access function value increases. This incentivises placing drop boxes near populations with low levels of access to other voting pathways.

While it is desirable to increase voter turnout and access to the voting system, expanding the use of ballot drop boxes may increase the financial cost of managing the election. The costs of the ballot drop box system can be broken into two major groups: fixed or operational. *Fixed costs* represent the "per drop box" costs such as the initial purchase and costs of securing and maintaining the drop box. Each location may have a different fixed cost due to varying installation and security equipment requirements. Once drop boxes are installed, jurisdictions incur an *operational cost* for a bipartisan team to collect ballots from the drop boxes [99]. The operational cost is determined, in part, by the distance between drop boxes, the opportunity cost of bipartisan team's time, and the frequency at which the ballots are collected during an election. We assume that a bipartisan team collects ballots from all drop boxes whenever a collection is conducted, and the drop boxes are visited in an order that minimizes the operational cost, referred to as the *collection tour*.

In addition to introducing new financial costs, ballot drop boxes introduce three types of risks to the voting process that can be mitigated through design requirements. The first risk is that ballot drop boxes can be misallocated in a way that causes voter suppression [163]. There are two components to this risk. The first is the potential to misallocate drop boxes such that access to the drop box voting system is inequitable. The second is the potential to misallocate drop boxes such that the access to the entire voting system defined by the multiple voting pathways is inequitable. These risks are reflected by the number of voters covered by a drop box, using the same definition of coverage introduced earlier, and the value of the access function for each voter, respectively. We can mitigate the risk of voter suppression by requiring that each voter is covered by at least one drop box and that the value of the access function meets some minimal threshold for all voters.

The second risk is that a drop box could be damaged or destroyed [163]. A nefarious actor could influence an election by targeting drop boxes that provide access to certain voters. The impact of this risk can be mitigated by requiring all voters to be covered by multiple drop boxes, so that voters have redundant access to the drop box system.

The last risk is that ballots submitted to a drop box could be stolen or manipulated. The impact of this risk can be mitigated by ensuring that the collection tour has a low cost. When the collection tour has a low cost, election officials can collect ballots often, leaving fewer at risk. Other implicit design choices also mitigate this third risk. For example, requiring a bipartisan team to collect ballots, rather than one individual, reduces the risk of an insider attack. Likewise, incorporating security related costs, such as the cost of a video surveillance system, into the fixed cost of a drop box mitigates the risks associated with it.

There are additional risks and mitigations associated with the voting process that are not unique to the drop box infrastructure. For example, there is a risk of an insider attack on ballots stored at an election building after being collected from the drop boxes [163]. However, these additional risks are outside the scope of the drop box system considered in this chapter (see Figure 4.1).

4.3.2 The Drop Box Location Problem (DBLP)

We now formally introduce an IP formulation of the drop box location problem (DBLP) using the sets, parameters, and variables presented in Table 4.1.

The DBLP selects drop box locations from a set of potential locations, N. Potential drop box locations can be identified using existing guidelines [99, 122]. Let y_n be a decision variable that equals one if a drop box is located at location $n \in N$ and zero otherwise. Once drop box locations are selected, a collection tour over them must be found to determine the operational cost of the drop box system. Let x_{ij} be a decision variable that equals one if the collection tour travels between drop box i and drop box j, $(i, j) \in E$, and zero otherwise,

Table 4.1: Notation

Sets

W	=	the set of voter populations
N	=	the potential drop box locations
$T \subseteq N$	=	the locations at which a drop box must be placed
E	=	all pairs $i \in N$, $j \in N$ such that $i \neq j$ and $(j, i) \neq E$
$N_w \subseteq N$	=	drop box location that cover $w \in W$, $\lvert N_w \rvert \geq 2$

Parameters

s	=	the start and end of the collection tour
f_j	=	the fixed cost of placing a drop box at location $j \in N$
c_{ij}	=	the operational cost of traveling between $i \in N$ and $j \in N$ in the collection tour
v_w^0	=	the propensity of $w \in W$ not to vote
v_w^1	=	the accessibility of the non-drop box voting system to $w \in W$
a_{jw}	=	the accessibility of location $j \in N$ to $w \in W$
r	=	minimal allowable value for the access function
q	=	minimal number of drop boxes covering each $w \in W$

Decision Variables

x_{ij}	=	1 if the collection tour moves between i and j $(i, j) \in E$ and 0 otherwise
y_j	=	1 if a drop box is placed at location $j \in N$ and 0 otherwise

where E represents all pairs $i \in N$, $j \in N$ such that $i \neq j$ and $(j, i) \neq E$. We assume the collection tour always begins and ends at a drop box[7] located at s (e.g., primary municipal building). Let T represent the locations at which there must be a drop box within our solution (e.g., existing drop box locations). The set T is always non-empty, since $T = \{s\}$ in the extreme case. For each location $j \in N$, let f_j equal the fixed cost of a drop box at j. Let c_{ij} represent the operational cost of traveling between drop boxes $(i, j) \in E$ on the collection tour.

Using this notation, we formalize the three goals of the DBLP. The first goal is to minimize the total cost associated with the selected drop box locations. The total cost of the drop box system is the sum of the fixed costs and the cost of the collection tour,

[7]When a drop box is not located at s, then the model is still valid. Simply let $f_s = 0$ and $a_{sw} = 0$ for each $w \in W$, while s is not a member of any covering set.

$z_1 := \sum_{j \in N} f_j y_j + \sum_{(i,j) \in E} c_{ij} x_{ij}$. The value of z_1 serves as the objective[8] function in the IP formulation of the DBLP.

The second goal of the DBLP is to equitably improve the accessibility of the voting system. Let W denote the collection of voter populations. Let $N_w \subseteq N$ represent the drop boxes that cover $w \in W$. We ensure equitable access to the drop box system[9] by requiring that each voter is covered by q drop boxes. Reasonable values of q are 0, 1, or 2. The cardinality of each covering set must be at least q, $|N_w| \geq q$ for all $w \in W$, otherwise the problem is infeasible. We ensure equitable access to all voting pathways by requiring that the access function value is at least r for each $w \in W$, $\min_{w \in W} A_w(N') \geq r$ where $N' = \{n \in N : y_n = 1\}$ are the selected drop box locations. This constraint can be viewed as a second objective for the DBLP using the epsilon-constraint approach for multi-objective optimization problems [120].

The third goal of the DBLP is to mitigate the risks associated with the drop box voting system. The risk of misallocating drop boxes in a way that leads to voter suppression is addressed by the second goal of the DBLP. The risk of ballots being susceptible to manipulation once submitted to a drop box is addressed by minimizing the cost of the collection tour, which is captured within z_1. The risk of damage to or destruction of drop boxes is a way that degradates voter access to the voting system is mitigated by ensuring each voter is covered by q drop boxes when $q \geq 2$.

If the optimal solution to the DBLP locates two or fewer drop boxes[10], the collection tour visiting the drop box locations is trivial. Thus, we assume that at least three drop boxes are be selected in the optimal solution. Under this assumption, we can formulate the

[8] Election administers likely have a fixed budget, but the amount allocated to managing the drop box system is likely not predetermined. Thus, we wish to minimize the proportion of the budget allocated to the drop box system.

[9] When $q \geq 1$.

[10] It can be easily checked whether two or fewer drop boxes are needed to satisfy the constraints of the model.

DBLP using the following IP.

$$\min_{x,y} z_1 = \sum_{(i,j)\in E} c_{ij}x_{ij} + \sum_{j\in N} f_j y_j \tag{4.1}$$

$$\text{s.t. } r(v_w^0 + v_w^1 + \sum_{j\in N} a_{jw}y_j) \leq v_w^1 + \sum_{j\in N} a_{jw}y_j \qquad \forall\, w \in W \tag{4.2}$$

$$\sum_{j\in N_w} y_j \geq q \qquad \forall\, w \in W \tag{4.3}$$

$$y_j = 1 \qquad \forall\, j \in T \tag{4.4}$$

$$\sum_{i\in N:(i,j)\in E} x_{ij} = 2y_j \qquad \forall\, j \in N \tag{4.5}$$

$$\sum_{\substack{(i,j)\in E: i\in S, j\in N\setminus S \\ \text{or } j\in S, i\in N\setminus S}} x_{ij} \geq 2y_t \qquad \forall\ S\subset N,\ 2\leq|S|\leq|N|-2,\ T\setminus S\neq\emptyset, t\in S \tag{4.6}$$

$$y_j \in \{0,1\} \qquad \forall\, j \in N \tag{4.7}$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in E \tag{4.8}$$

The objective (4.1) is to minimize the total cost of the drop box system. Constraint set (4.2) requires that the value of the access function is at least r for each $w \in W$. Constraint set (4.3) ensures that each $w \in W$ is covered by at least q drop boxes within their respective covering set. Constraint set (4.4) ensures that a drop box is added at each location in T. Constraint sets (4.5) and (4.6) are used to determine the collection tour over the selected drop box locations using constraints originally introduced for the CTP [78]. Constraint set (4.5) ensures that each selected drop box location is visited by the collection tour exactly once. Constraint set (4.6) introduces subtour elimination constraints. Note that these constraints differ from the subtour elimination constraints commonly seen in the TSP, since not all locations N must be visited by collection tour. The bound on the summation refers to the edges in E such that the edge is incident to one node in S and one in $N \setminus S$. Constraint sets (4.7) and (4.8) require the decision variables to be binary.

4.3.3 Model Properties

The DBLP is challenging to solve using existing solution techniques. This idea is formalized in Theorem 4.1, which states that the DBLP is NP-Hard. A proof is provided in Appendix 7.

Theorem 4.1. *The DBLP is NP-Hard.*

In some instances, the DBLP integer program may be large due to a large number of voter populations. We present a condition that allow us to determine when certain constraints from constraint set (4.2) can be removed from the IP, which reduces the size of the integer program instance and potentially reduces the time needed to find an optimal solution. Lemma 1 gives a sufficient condition for which the constraint corresponding to a voter population $w \in W$ in constraint set (4.2) can be removed from the DBLP integer program, since the access function value is guaranteed to be smaller for another voter population $\hat{w} \in W$ for all choices of drop box locations. A proof is provided in Appendix 7.

Lemma 1. *Let $w, \hat{w} \in W$ be two voter populations. If the access function parameters satisfy $v^0_{\hat{w}} \geq v^0_w$, $v^1_{\hat{w}} + \sum_{n \in T} a_{n\hat{w}} \leq v^1_w + \sum_{n \in T} a_{nw}$, and $a_{n\hat{w}} \leq a_{nw}$ for each $n \in N \setminus T$, then for any subset of drop box locations $N' \subseteq N$, such that $T \subseteq N'$:*

$$\frac{v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}}{v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}} \leq \frac{v^1_w + \sum_{n \in N'} a_{nw}}{v^0_w + v^1_w + \sum_{n \in N'} a_{nw}}$$

This property may be satisfied in realistic instances of the DBLP. Consider population $\hat{w}$ that lies on the exterior boundary of the jurisdiction. Consider a w that lies just inside of $\hat{w}$ within the jurisdiction such that w is closer than $\hat{w}$ to all potential drop box locations and polling locations. The voters in w have higher access to the voting infrastructure than the voters in $\hat{w}$. In this case, the properties of Lemmas 1 are likely to be satisfied.

4.3.4 Model Variations

The DBLP is designed to determine drop box locations that satisfy the requirements for a drop box system in most jurisdictions. However, election administrators may wish to explore solutions that are not identified by the standard formulation of the DBLP or may wish to tailor the model to their situation. In this subsection, we discuss five modifications that can be made to the DBLP.

First, the value of q determines the number of drop boxes that must cover each voter, and determines the access that the voters have to the drop box system to an extent. Letting $q = 0$, voters are not required to be covered by drop boxes. Instead, a cost effective set of drop box locations are selected such that all voters have a minimum level of access (r) to the voting system. Letting $q = 1$, drop box locations are selected so that each voter is guaranteed access to the drop box system in addition to meeting a minimal level of access to all voting pathways (r). Letting $q = 2$, voters are guaranteed access to both the drop box and complete voting system in a way that also mitigates the impact the destruction of a drop box could have on voter access.

The second variation we consider is a change to the covering sets N_w for $w \in W$. The covering sets N_w typically include locations within a predefined time or distance threshold from a voter. Decreasing or increasing the time threshold used can make constraint set (4.3) more or less restrictive, respectively. When the covering sets are determined using a shorter time threshold, drop boxes within a voter's covering set are required to be located closer to the voter. This may make the drop boxes more accessible to all voters, but also increases costs. When the covering sets are determined using a larger time threshold, drop boxes are allowed to be located further away while still satisfying constraint set (4.3), which results in lower cost.

Third, we can replace the cost objective of the DBLP with other goals. We can instead

maximize the minimum access function[11] or maximize the number of voters covered by at least q drop boxes. In the latter case, we introduce a new indicator variable δ_w and the following constraint

$$q\delta_w \leq \sum_{j \in N_w} y_j \quad \forall\, w \in W \tag{4.9}$$

The objective is then to maximize $\sum_{w \in W} p_w \delta_w$ where p_w represents the number of voters in w. With this objective, we can use constraint set (4.3) to ensure each population $w \in W$ is covered by at least some q' $(0 \leq q' < q)$ drop boxes. When z_1 is no longer the objective of the integer program, a constraint can be added to ensure that the cost of the drop box system is no more than some budget B,

$$\sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{n \in N} f_n y_n \leq B \tag{4.10}$$

With this constraint, feasibility of the DBLP is no longer guaranteed. Infeasibility can be informative to election administrators.

Fourth, election administrators may wish to restrict the cost of the collection tour, since they may have a limited operational budget for collecting ballots (e.g., limited staff). We can limit the cost of the collection tour to no more than $c_{\max}$ by introducing the following constraint

$$\sum_{(i,j) \in E} c_{ij} x_{ij} \leq c_{\max} \tag{4.11}$$

Fifth, election administrators may wish to locate a specific number, k, of drop boxes. This may occur when they have already purchased drop boxes or they wish to add a drop

[11] It is fairly straightforward to convert constraint set (4.2) into a linear equivalent by using one minus the access function value, which is an equivalent measure of access.

box for every 15,000-20,000 registered voters as recommended by [99]. This can be enforced by adding the following constraint

$$\sum_{n \in N} y_n = k \tag{4.12}$$

4.4 Solution Methods

Within this section, we present solutions methods for the original DBLP formulation.

4.4.1 Objective Reformulation

Constraint sets (4.3)-(4.6) are similar to constraints that may be found in an integer program for the CTP [78]. However, the objective of the CTP only considers operational costs [78]. Thus, it is desirable to reformulate objective z_1 to preserve properties of the CTP within the DBLP. We can then use components of solution methods for the CTP within solution methods for the DBLP.

We present a reformulation of z_1 to remove the use of y_n variables. Note that constraints (6) enforce that for any drop box n visited by a feasible tour, there must be exactly two drop boxes visited before or after n. Thus, we can reformulate z_1 as follows:

$$\begin{aligned} z_1 &= \sum_{i,j \in E} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j \\ &= \sum_{i,j \in E} c_{ij} x_{ij} + \sum_{j \in N} \sum_{i \in N : i,j \in E} f_j x_{ij}/2 \\ &= \sum_{i,j \in E} (c_{ij} + f_i/2 + f_j/2) x_{ij} \\ &= \sum_{i,j \in E} \hat{c}_{ij} x_{ij} \end{aligned}$$

where $\hat{c}_{ij} := c_{ij} + f_i/2 + f_j/2$ for each $(i, j) \in E$. With this reformulation, z_1 takes the same

form as the standard objective for the CTP.

4.4.2 Lazy Constraint Method

Branch and bound is one of the most common techniques used to solve IPs, and we employ it to solve the DBLP. However, constraint set (4.6) defines an exponential number of constraints, so we introduce a lazy constraint approach to solve the DBLP. First, we solve the DBLP without constraint set (4.6). Once an optimal solution is found, we determine if any of the constraints from constraint set (4.6) are violated. If so, we add in at least one violated constraint and resolve the IP using branch and bound. Most modern optimization packages support the implementation of lazy constraints. The reformulation of the objective introduced in Section 4.4.1 can be used throughout the procedure, but it is not required.

We introduce a new polynomial time algorithm, Algorithm 2, to find violated inequalities from constraint set (4.6) given an $x^* \in \{0,1\}^{|E|}$. The approach we take is adapted from an approach used for the TSP [85] to account for the fact that not all potential drop box locations must be visited by the tour in the DBLP. Algorithm 2 first finds all subtours defined by x^* (line 1). Each subtour that does not include all required locations T (lines 2-4) must be associated with least one violated constraint. For all[12] locations t visited by the subtour, we add the violated constraint (line 7).

Algorithm 2 Lazy(x^*)

1: $H =$ collection of subtours defined by x^*
2: **for** each subtour $h \in H$ **do**
3: $\quad \hat{S} =$ drop box locations visited by h
4: $\quad$ **if** $T \setminus \hat{S} \neq \emptyset$ **then**
5: $\quad\quad$ **return** $\sum_{(i,j)\in E: i\in\hat{S}, j\in N\setminus\hat{S} \text{ or } j\in\hat{S}, i\in N\setminus\hat{S}} x_{ij} \geq 2y_t$ for each node $t \in \hat{S}$
6: $\quad$ **end if**
7: **end for**

[12]There is a trade-off between adding a constraint for all $t \in \hat{S}$ (increasing the size of the IP) and adding a constraint for a small number of elements in $\hat{S}$ (increasing the number of times the search procedure occurs).

We comment on the correctness of Algorithm 2. Specifically, given an integer $x^* \in \{0,1\}^{|E|}$, Algorithm 2 finds a violated constraint from constraint set (4.6), if one exists. If a constraint is violated, there must exist a S such that $S \subset N,\ 2 \leq |S| \leq |N| - 2,\ T \setminus S \neq \emptyset$ and for some $t^* \in S$, $\sum_{(i,j)\in E: i\in\hat{S}, j\in N\setminus\hat{S} \text{ or } j\in\hat{S}, i\in N\setminus\hat{S}} x^*_{ij} < 2y_{t^*}$. Since the left hand side of the inequality is at least zero, t^* must represent a selected drop box location ($y_{t^*} = 1$). Moreover, the feasibility of x^* with regards to constraint set (4.5) implies that $\sum_{(i,j)\in E: i\in\hat{S}, j\in N\setminus\hat{S} \text{ or } j\in\hat{S}, i\in N\setminus\hat{S}} x^*_{ij} = 0$. Thus, t^* must be a member of some subtour visiting locations $\hat{S} \subseteq S$. The set $\hat{S}$ must contain at least three elements and can contain not more than $|N|-3$ elements as a result of constraint set (4.5). Since $T\setminus S \neq \emptyset$, it is also true that $T\setminus\hat{S} \neq \emptyset$. Thus, the existence of a S implies the existence of a $\hat{S}$ whose elements form a subtour in x^* such that $\hat{S} \subset N,\ 2 \leq |\hat{S}| \leq |N| - 2,\ T \setminus \hat{S} \neq \emptyset$ and $\sum_{(i,j)\in E: i\in\hat{S}, j\in N\setminus\hat{S} \text{ or } j\in\hat{S}, i\in N\setminus\hat{S}} x^*_{ij} < 2y_t$ for all $t \in \hat{S}$. Algorithm 2 identifies $\hat{S}$ and returns the corresponding constraint.

4.4.3 A Heuristic Method

The lazy constraint method can be used to find solutions to moderately sized problem instances, but large instances require an unreasonable amount of computational time. Moreover, if an appropriate value for r is not known by election administrators, the DBLP must be solved multiple times to allow election administrators to select among possible solutions a posteriori[13], which substantially increases the necessary computational time. In this section, we present a heuristic that identifies multiple solutions to the DBLP, each corresponding to a unique value of r. Depending on the implementation, the heuristic is polynomially solvable. We provide pseudocode for this heuristic in Appendix 9.1. The heuristic requires the objective reformulation discussed in Section 4.4.1.

The heuristic first identifies a feasible solution to the DBLP corresponding to $r = 0$.

[13]This may be desirable even if r is believed to be known.

When $q = 0$, we find a tour visiting the nodes of T using any method for the TSP[14]. When $q = 1$, the DBLP with $r = 0$ is equivalent to the CTP when the DBLP objective is reformulated as introduced in Section 4.4.1. Any solution method for the CTP can be used to identify a feasible solution. When $q = 2$, the DBLP with $r = 0$ is equivalent to the CTP when the DBLP objective is reformulated as introduced in Section 4.4.1, except the CTP only requires single coverage of each $w \in W$. We construct a solution that satisfies double coverage using any exact or heuristic solution method for the CTP as follows. First, we find a feasible solution to the CTP that ensures each $w \in W$ is covered once. Using this solution, we construct a second instance to the CTP. The second instance is equivalent to the first *except* that (1) the new set of required drop box locations includes all locations selected in the first solution, (2) the locations selected in the first solution are removed from all covering sets N_w, and (3) any $w \in W$ that was covered by at least two locations selected in the first solution is removed from W. A solution to the second CTP instance is guaranteed to be feasible for the DBLP when $r = 0$ and $q = 2$. A proof of this statement is provided in Appendix 7. If q takes a value greater than 2, it is fairly trivial to extend the process used when $q = 2$.

Once this initial solution corresponding to $r = 0$ is found, we wish to find solutions that are feasible for a larger r. These solutions can be found as follows. Initialize $r = 0$. We iterate and increase the value of r by some predetermined, sufficiently small ε in each iteration. Within an iteration, start with the solution identified by the previous iteration. Identify all pairs of drop box locations such that one is already included in the tour (not belonging to the set T) and the other is not. These pairs represent the locations that can be *swapped* (i.e., remove one from the current solution and add the other). We allow the pairs to also represent adding a location to the tour without removing another, or removing a location without adding another. The latter may be advantageous in cases where a drop

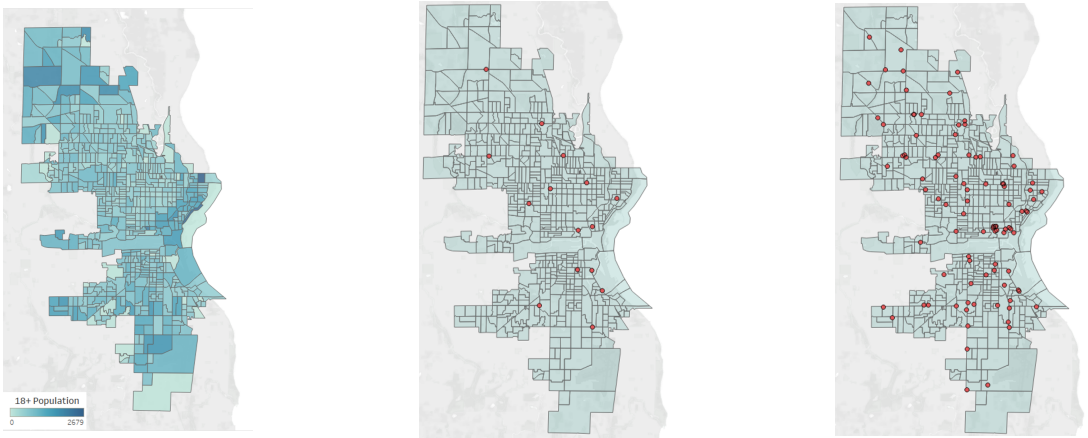
[14]Since we do not specify the methods to solve the TSP or CTP, this heuristic is in fact a *heuristic scheme*.

box location was added in a previous iteration that makes previously included locations redundant and unnecessary. There are $\mathcal{O}(|N|^2)$ possible pairs. We let a pair be *feasible* if the drop box locations obtained after the swap satisfy constraint set (4.2) according to the current value of r and satisfy the multiple coverage defined by constraint set (4.3). It is straightforward to check the feasibility of each pair. Note that we do not consider any pair that results in both an increased cost and lower minimal access function value, since it would directly lead to a dominated solution. Among all feasible pairs, determine the angle between the incumbent solution and the prospective solution, similar to what was done in [48]. Mathematical details can be found in Appendix 9.1. Select the pair that leads to the smallest angle; this incentivizes finding solutions with a lower cost. Then update the tour based on this pair in a cost minimizing way (e.g., minimum cost removal/insertion or other techniques used for the TSP [77]). Continue until all potential drop box locations have been selected in the solution. This is guaranteed to occur after a finite amount of time since r strictly increases by a fixed amount each iteration and is bounded above by one. Among all identified solutions, disregard the dominated solutions and return the rest. It can be checked during each iteration whether each new solution is dominated by a previous solution or if the new solution dominates a previous solution. When a polynomial time heuristic is used for the TSP and CTP, this heuristic also runs in polynomial time.

4.5 Case Study

We construct a case study of Milwaukee, Wisconsin to demonstrate the value of the DBLP and investigate the implications of optimal drop box infrastructure design. The City of Milwaukee is the most populous municipality in the state of Wisconsin and had approximately 315,483 registered voters prior to the 2020 General election [38]. We let W be defined by the census block groups of Milwaukee, WI [124], which are comprised of individuals

located near each other who are typically of similar socioeconomic backgrounds. Figure 4.2a illustrates the different block group locations in Milwaukee and the estimated number of individuals of age 18 or older in each [204].



(a) Milwaukee block group locations (W)

(b) Drop box locations in 2020

(c) Potential drop box locations (N)

Figure 4.2: (a) The census block groups of Milwaukee with a darker color indicating a higher population 18 years of age or older. (b) The drop box locations (red) used in 2020. (c) Potential drop box locations (red) within the City of Milwaukee used for this case study.

During the 2020 elections, 15 drop boxes were placed throughout Milwakuee [125], illustrated in Figure 4.2b. We use the DBLP to identify drop box locations assuming that these 15 were not already added to the voting system. This allows us to compare the DBLP to the decisions actually made by election officials during 2020. We let the potential drop box locations, N, be the locations of courthouses (4), election administrative buildings (2), fire stations (30), libraries (14), police stations (7), CVS pharmacies (7), and Walgreens pharmacies (29). Figure 4.2c illustrates the locations of the 93 potential drop box locations. We assume that the collection tour begins and ends at the Milwaukee City Hall, s. We do not require a drop box be located at any location other than City Hall, with $T = \{s\}$. The fixed cost of locating a drop box at court houses, fire stations, police stations, and City Hall is set at \$6,000 to reflect the cost of a drop box without the need of additional security

measures. The fixed cost of locating a drop box at all other locations is set at $10,000 to reflect the cost of both a drop box and a security system.

According to the Milwaukee Election Commission, ballots were, at a minimum, collected daily by staff during the 2020 General election [125]. This equates to approximately 50 times during the election. Based on this value, we assume that ballots are collected 50 times per year on average[15] over the life of the drop boxes, which we assume to be 15 years. We further assume that each member of the bipartisan collection team has an opportunity cost of $\$40$ per hour. This may not reflect the actual pay rate of poll workers or staff; rather, it is meant to represent the opportunity cost of other tasks not completed during that time. For example, staff could otherwise participate in additional security training, review compliance of submitted ballots, or conduct marketing to increase voter turnout. The cost of traveling between two drop boxes is determined using this pay rate and the estimated time needed to drive between the two locations, which is obtained from Bing Maps. We include the cost of gas and vehicle wear using the current federal mileage reimbursement of $\$0.56$ per mile. The estimated mileage is calculated assuming an average travel speed of 30 mph. Lastly, we assume the collection costs increase by two percent each year.

The covering set of each location, N_w, is constructed to include the locations that satisfy at least two of the following: the time to walk to the drop box is no more than 15 minutes; the time to drive to the drop box is no more than 15 minutes; the time to use public transit (i.e., city bus) to the drop box is no more than 30 minutes; or the road distance to the drop box is no more than 4 miles (e.g., reachable by bike or ride-share). By ensuring at least two conditions are are met, there must be multiple transportation modalities that can be used to travel to a drop box in N_w. Individuals without access to a private vehicle are thereby guaranteed to be able to reach a covering drop box using another mode of transportation.

[15] In reality, the number of collections depends on the election year. Also, the frequency of ballot collection may vary depending on model solutions, but this value is set to normalize the operational cost to the fixed cost of the drop boxes.

We estimate the location of each $w \in W$ using the block group centriod. Throughout the case study, we let $q = 2$, unless otherwise noted, so that model solutions mitigate the risk associated with the destruction of a drop box.

Lastly, the parameters $v_w^0,\ v_w^1,\ a_{nw}$ for the access function are instantiated as follows. Ideally, these parameters would be determined using a multinomial logistic regression based on surveys, distance to voting locations, and available transportation methods. Due to a lack of available data, we introduce a function that serves as a proxy. Our function combines historical voter turnout, transit durations obtained from Bing Maps, vehicle availability of individuals living in each block group [203], and the work locations of individuals residing each block group [201]. We let v_w^1 equal the estimated voter turnout (between 0 and 1) among registered voters in each block group during the 2016 General election, and we let $v_w^0 = 1 - v_w^1$. The values reflect the idea that voter turnout is higher where the in-person voting system is more accessible [26]. To describe the values of a_{nw}, we introduce some notation. Let $d_{n,w}^{\text{walk}}$, $d_{n,w}^{\text{transit}}$, $d_{n,w}^{\text{drive}}$ be the walking, transit, and driving durations (minutes), respectively, to potential location $n \in N$ for population $w \in W$ obtained from Bing Maps. Let $\lambda_w^{\text{vehicle}}$ be the proportion of individuals in w that have access to a vehicle according to the U.S. Census [203]. Let $d_{n,w}^{\text{other}}$ be the estimated duration to the potential drop box location $n \in N$ from population $w \in W$ using some other form of transportation (e.g., bike or rideshare); the duration is calculated using the road distance obtained from Bing Maps and an assumed speed of 15 miles per hour. Let Q represent the set of work locations in Milwaukee, WI [201]. Let $d_{n,q}^{\text{work}}$ be the walking duration from work location $q \in Q$ to the potential drop box location $n \in N$. Let $\lambda_{w,q}^{\text{work}}$ be the proportion of individuals from w that work in location q according to the U.S. Census [201]. Then, the value of a_{nw} for a drop box location n and population $w \in W$ is computed as:

$$a_{nw} = \frac{0.04}{v_w^1}\left(\frac{1}{(d_{n,w}^{\text{walk}})^2} + \frac{1}{(d_{n,w}^{\text{transit}})^2} + \frac{\lambda_w^{\text{vehicle}}}{(d_{n,w}^{\text{drive}})^2} + \frac{1}{(d_{n,w}^{\text{other}})^2} + \sum_{q \in Q} \frac{\lambda_{w,q}^{\text{work}}}{(d_{n,q}^{\text{work}})^2}\right)$$

This formula accounts for the benefit of multiple modes of transportation to a location and assumes that drop boxes near individuals are much more accessible than drop boxes far away. We include the term $(v_w^1)^{-1}$ to account for the added intangible benefit of drop boxes located near a population with historically low voter turnout, such as increased publicity and visual reminders to cast a ballot [40]. We square the duration of each transportation mode to model a non-linear relationship between duration and accessibility (a similar approach was employed in [138]). As a result, a_{nw} is highest when the drop box is a short duration from the voter using each mode of transportation. We scale each a_{nw} by 0.04 to produce values that align with findings from empirical research [122]. Additional explanation and justification of this function is provided in Appendix 9.2.

4.5.1 The DBLP and Rules-of-thumb

In the absence of tools to support election planning, election administrators may use rules-of-thumb to select drop box locations. In this section, we demonstrate that the DBLP is able to identify drop box locations that outperform rules-of-thumb across multiple criteria. The findings support the use of the DBLP during election planning. Table 4.2 presents the details of DBLP solutions for different values of r obtained using the Gurobi 9.1 MIP solver. Computational studies were run using 64 bit Python 3.7.7 on an Intel® Core™ i5-7500 CPU with 16 GB of RAM. Each optimal solution was identified in less than 3600 seconds. We refer to the solutions identified by the DBLP as 'DBLP k' where k refers to the solution ID in Table 4.2.

During 2020, the Milwaukee Election Commission located drop boxes at the City Hall, the Election Commission warehouse, and 13 neighborhood-based public library branches [125]. We begin by comparing these locations to those identified in DBLP 2, which also represents 15 drop boxes. Table 4.3 provides the values of multiple criteria for each drop box system. These criteria provide insight into the performance of each drop box system

Table 4.2: Properties of solutions obtained by solving the DBLP with different values of r.

Solution ID	Minimum Access Function Value	Tour Cost ($/yr)	Fixed Cost ($/yr)	Operational cost ($/yr)	Number of Drop Boxes
0*	0.573	17,141	6,800	10,341	15
1	0.582	17,313	6,800	10,513	15
2	0.593	17,813	7,333	10,479	15
3*	0.601	18,538	8,267	10,271	16
4	0.612	19,701	8,800	10,901	18
5	0.620	21,697	10,267	11,430	21
6*	0.629	23,599	12,133	11,466	23
7	0.637	26,877	14,133	12,743	28
8	0.645	31,701	17,200	14,501	33
9	0.653	37,318	20,933	16,385	39
10	0.661	47,023	27,200	19,823	52
11	0.668	80,574	50,800	29,774	93

* Illustrated in Figure 4.5

with respect to cost, access, and risk.

The results in Table 4.3 suggests that the DBLP is able to identify drop box locations that perform better across multiple criteria compared to the rule-of-thumb used by election administrators in Milwaukee during the 2020 election. We find that with the same number of drop boxes, the DBLP is able to identify drop box locations that result in a lower fixed cost, operational cost, and total cost. Despite having a lower cost, all voters are covered by at least two drop boxes with DBLP 2, while the 2020 policy only covers 88.9% of voters twice. This gap also exists when voters do not have access to a vehicle (1.00 vs. 0.941). This means that DBLP 2 admits a higher level of equity of access to the drop box system while mitigating the risk associated with the possible destruction of a drop box. Moreover, the minimum access function value is higher (0.593 vs. 0.560) for DBLP 2. With a strict interpretation of the access function, the block group with the lowest turnout is predicted to have a turnout that is 3.3% higher if the DBLP 2 was implemented rather than the actual locations. We find that the average access function value is lower for DBLP 2 than the

Table 4.3: Comparison of the actual 2020 drop box system to a drop box system design identified by the DBLP across multiple criteria.

Criteria	2020	DBLP 2
Number of Drop Boxes	15	15
Fixed Cost ($/year)	9,733	7,333
Operational cost ($/year)	10,566	10,479
Total Cost ($/year)*	20,300	17,813
Fraction of voters covered by 1 drop box (population weighted)	0.995	1.000
Fraction of voters covered by 2 drop boxes (population weighted)†	0.889	1.000
Fraction of voters without access to a car covered by at least two drop boxes by non-driving transit (population weighted)‡	0.941	1.000
Minimum access function value§	0.560	0.593
Average access function value (population weighted)	0.776	0.772
Maximum road distance to closest drop box	7.634	6.311
Maximum road distance to third closest drop box	10.55	9.978
Average road distance to closest drop box (population weighted)	1.601	1.679
Average road distance to closest 3 drop boxes (population weighted)	2.723	2.486

* Objective of the DBLP.

† Required by constraint set (4.3) of the DBLP.

‡ Required by constraint set (4.3) given our method of instantiating N_w for each $w \in W$.

§ Modeled using constraint set (4.2) in DBLP.

actual implementation; however, the difference is small (0.772 vs. 0.776).

In different situations, other rules-of-thumb may be used by election administrators. We compare the DBLP solutions to six other rules-of-thumb that could have potentially been used instead:

Policy 1. Locate drop boxes at the election administrative buildings (2).

Policy 2. Locate drop boxes at the election administrative buildings (2) and police stations (7).

Policy 3. Locate drop boxes at the election administrative buildings (2) and libraries (14).

Policy 4. Locate drop boxes at the election administrative buildings (2), police stations (7), and libraries (14).

Policy 5. Locate drop boxes at the election administrative buildings (2) and fire stations (30).

Policy 6. Locate drop boxes at the election administrative buildings (2), police stations (7), libraries (14), and fire stations (30).

We choose to assess these policies, since they represent the placement of drop boxes at buildings that should be well-distributed throughout the city. They are not intended to represent a comprehensive list of possible policies. Table 4.4 provides the values of multiple criteria for each rule-of-thumb and DBLP solutions with a similar number of drop boxes. The results presented in Table 4.4 mirror the findings reported in Table 4.3; the DBLP identifies drop box locations that are consistently better across multiple criteria than rules-of-thumb with a similar number of drop boxes. Moreover, most rules-of-thumb are not feasible for the DBLP. Policy 6, which locates 53 drop boxes, is the only rule-of-thumb policy that guarantees that each $w \in W$ is covered by $q = 2$ drop box locations. Meanwhile, the DBLP can find feasible solutions with as few as 15 drop boxes.

Table 4.4: A comparison of rule-of-thumb and DBLP policies across multiple criteria.

Criteria	Policy 1	Policy 2	Policy 3	Policy 4	Policy 5	Policy 6	DBLP 3	DBLP 6	DBLP 8	DBLP 10
Number of Drop Boxes	2	9	16	23	32	53	16	23	33	52
Fixed Cost ($/year)	1,067	3,867	10,400	13,200	13,067	25,200	8,267	12,133	17,200	27,200
Operational cost ($/year)	1,535	7,233	10,616	11,954	18,328	21,129	10,271	11,466	14,501	19,823
Total Cost ($/year)*	2,602	11,100	21,016	25,154	31,395	46,329	18,538	23,599	31,701	47,023
Fraction of voters covered by 1 drop box (population weighted)	0.596	0.972	0.995	0.995	1.000	1.000	1.000	1.000	1.000	1.000
Fraction of voters covered by 2 drop boxes (population weighted)†	0.362	0.810	0.924	0.973	0.997	1.000	1.000	1.000	1.000	1.000
Fraction of voters without access to a car covered by at least two drop boxes by non-driving transit (population weighted)‡	0.467	0.920	0.963	0.981	1.000	1.000	1.000	1.000	1.000	1.000
Minimum access function value§	0.542	0.558	0.568	0.582	0.591	0.623	0.601	0.629	0.645	0.661
Average access function value (population weighted)	0.760	0.770	0.777	0.786	0.790	0.809	0.773	0.781	0.788	0.806
Maximum road distance to closest drop box	19.269	9.978	7.634	7.634	5.568	5.568	6.311	6.311	6.311	4.865
Maximum road distance to third closest drop box	20.123	14.199	10.554	9.978	8.567	8.249	9.851	8.653	8.249	8.249
Average road distance to closest drop box (population weighted)	5.829	2.130	1.550	1.469	1.062	0.917	1.689	1.517	1.386	1.105
Average road distance to closest 3 drop boxes (population weighted)	6.687	3.348	2.578	2.122	1.774	1.399	2.436	2.052	1.837	1.534

* Objective of the DBLP.

† Required by constraint set (4.3) of the DBLP.

‡ Required by constraint set (4.3) given our method of instantiating N_w for each $w \in W$.

§ Modeled using constraint set (4.2) in DBLP.

4.5.2 Drop Box Trade-offs

In this section, we further investigate DBLP solutions and explore the trade-offs between criteria within the drop box voting system. We begin by discussing the trade-off between cost and equity of access to all voting pathways (i.e., the minimum access function value). The solutions in Table 4.2 suggest there is a substantial trade-off between the cost of the drop box system and the minimum access function value. However, the marginal increase in cost to achieve an increase in the minimum access function value is not constant. From DBLP solution 0 to DBLP solution 1 the average cost of a 0.01 increase of the minimum access function value is $\$186.84$ per year. From solutions 3 to 4 the average cost of a 0.01 increase of the minimum access function value is $1,062.25 per year. From solutions 10 to 11 the average cost of a 0.01 increase of the minimum access function value is $50,738 per year. This highlights the importance of considering the access function within the DBLP. When a low cost solution is desirable, an appropriate value for r allows the DBLP to identify drop box locations that admit a much larger minimum access function value for a relatively low increase in cost (e.g., solutions 1-4). When drop boxes that admit a large minimum access function value are desirable, it is critical to appropriately set r, since a small change in r can lead to solutions of substantially different cost (e.g., solutions 8-10).

We next consider the trade-off between equitable access to the drop box system and equitable access to all voting pathways. Figure 4.3 plots the cost and minimum access function value of multiple optimal solutions when q is zero (—●—), one (—□—), or two (—△—) with the latter corresponding to the solutions presented Table 4.2. When $q = 0$, the DBLP is able to identify drop box locations that substantially increase the minimum access function value for a relatively small cost. This suggests that there are cost-effective, equitable drop box locations, even when election officials cannot afford to cover each voter with one or two drop boxes. In general, equitable access to the drop box system and equitable access

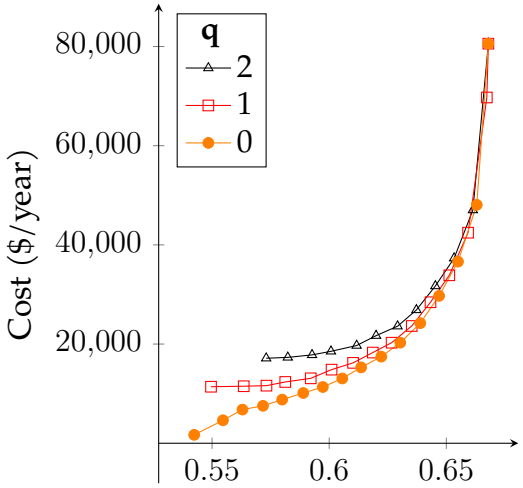


Figure 4.3: Solutions using different values of q.

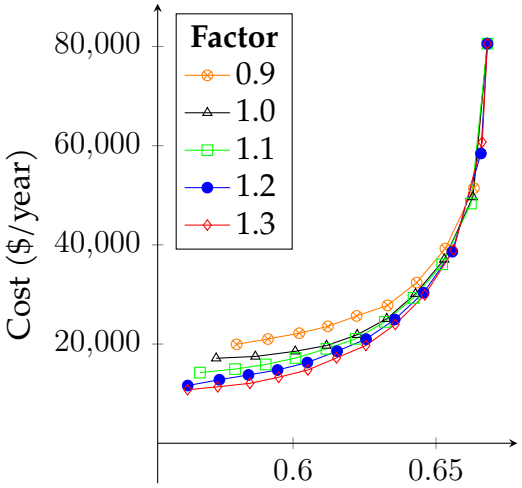


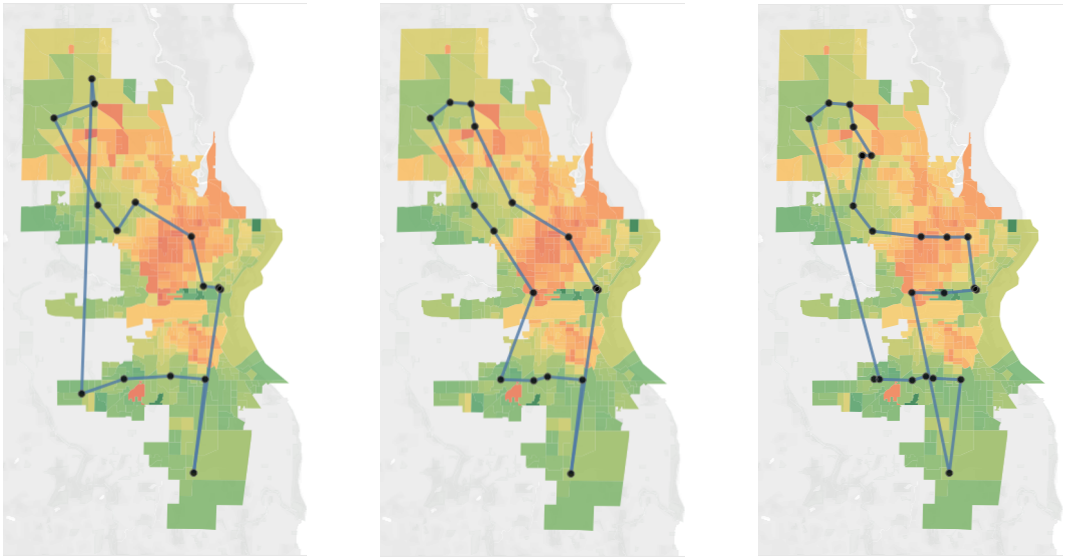
Figure 4.4: Solutions using different time thresholds for N_w determined by the factor.

to all voting pathways are aligned so that access is improved. However, the difference between the curves corresponding to $q = 0$ (—●—) and $q = 1$ (—□—) represents the cost of ensuring equitable access to the drop box system. In some cases, this cost can be substantial ($\sim \$6,767$ per year). This demonstrates the trade-off between selecting drop boxes that ensure all voters have access to the drop box system or using the drop boxes to increase the access function value by "filling in the gaps" of the in-person voting system. We also find a substantial difference between the curves corresponding to $q = 1$ (—□—) and $q = 2$ (—△—), particularly when a low cost solution is desired. This suggests that the cost of mitigating risks associated with the destruction of drop boxes through infrastructure design is relatively high and may not be cost effective. Instead, it may be more cost effective to respond to an adverse event after it occurs, since the likelihood of this risk occurring is low [163].

Rather than changing q, we can also relax coverage by defining the coverage sets N_w using a larger time threshold. When covering sets are defined by a longer time threshold, the drop boxes are allowed to be located further away from the voters while still meeting the coverage constraints defined in constraint set (4.3). A larger time threshold may increase the inequity of access to the drop box infrastructure within the resulting solutions, since a census block group may be further from both covering drop boxes when compared to other census block groups. However, the access function continues to evaluate the effect of the drop box locations on voter turnout when all voting pathways are considered. Figure 4.4 illustrates the cost and minimum access function value of solutions to the DLBP using covering sets N_w obtained using the same procedure as before, except the time threshold are multiplied by factors of 0.9 (⊗), 1.0 (△), 1.1 (□), 1.2 (●), and 1.3 (◇) when $q = 2$. A factor of 1.0 corresponds to the covering sets used to obtain the solutions discussed in Table 4.2 and Figure 4.3. We find the effects of changing the covering sets to be similar to the effect of changing q. Covering sets defined by a larger factor result in solutions with a larger minimum access function value for the same cost.

We explore solutions DBLP 0, 3, and 6 of Table 4.2 in more detail. Figure 4.5 illustrates the selected drop box locations and the collection tour visiting these drop boxes overlaid on a map of Milwaukee. Each black circle indicates a selected drop box location, and the blue lines describe the order in which the drop boxes are visited on the collection tour (not the actual roads driven). The color of each block group indicates the access function value of each block group with red reflecting relatively a low value and green reflecting relatively high value. When cost is minimized (Solution 0), drop boxes are well-spaced in order to cover each block group twice, but relatively few drop boxes are placed to reduce cost. Solution 0 is notably different than the locations selected during 2020, Figure 4.2b, in the northern and southern areas of the city despite locating the same number of drop boxes. The DBLP selects additional drop box locations in the north and south to ensure equitable

access to the drop box system within those region. When the cost and minimal access function value are higher (Solutions 3 and 6), more drop boxes are added. The DBLP selects additional drop box locations in the middle and northern part of the city, which would otherwise have have a relatively low level of access to the voting infrastructure, indicated by the dark red in Figure 4.5(a). Additional locations are not selected in the south, since those voters have relatively high access to the multiple voting pathways, indicated by the dark green in Figure 4.5(a).



(a) Solution 0 (b) Solution 3 (c) Solution 6

Figure 4.5: Optimal drop box locations and tour visiting these locations. Color of regions reflect the access function value (red is low, green is high).

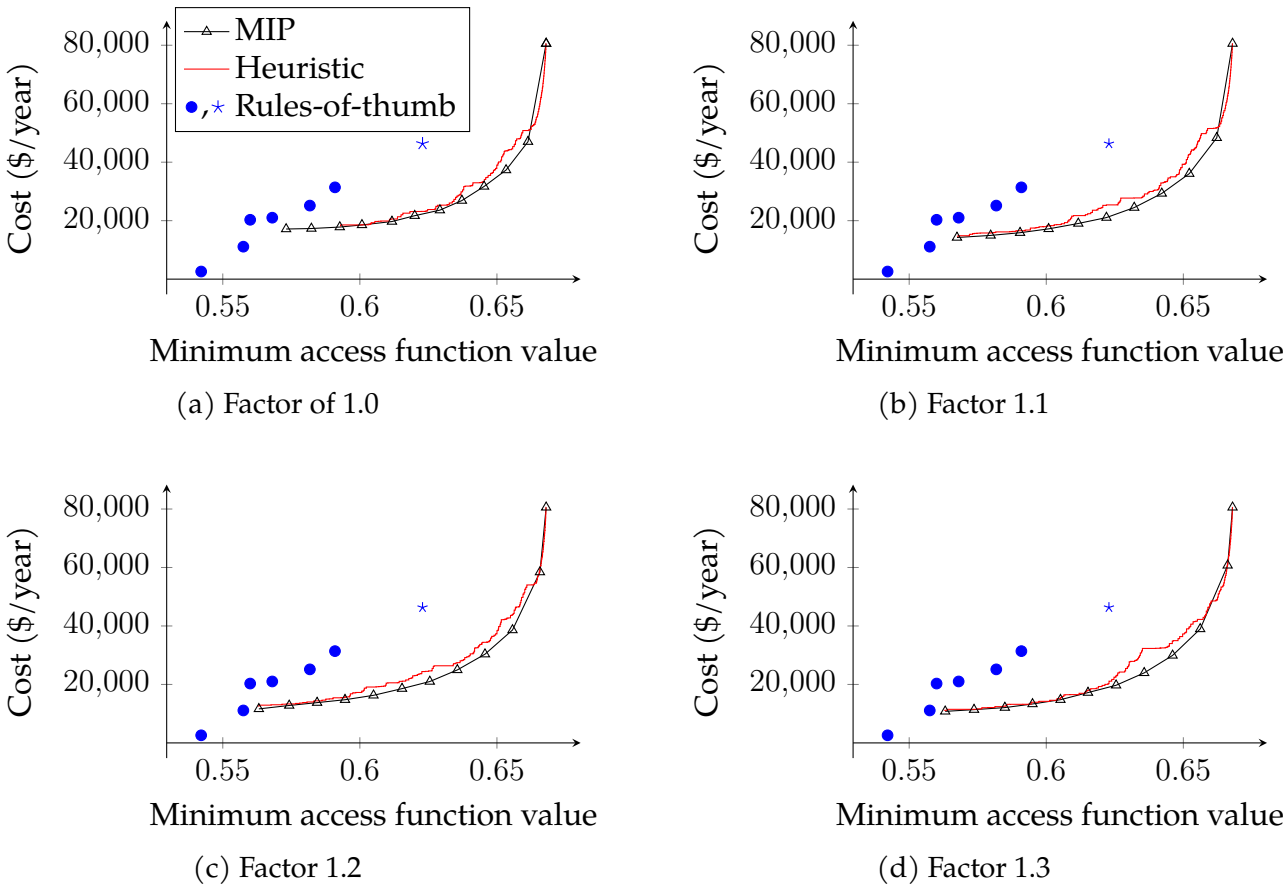
4.5.3 Heuristic Results

In this section, we investigate the performance of the DBLP heuristic compared to the lazy constraint approach. The lazy constraints were implemented using the Gurobi 9.1 MIP solver. Instances to the TSP and CTP created during the heuristic's execution were solved using the following methods. To solve a CTP instance, we formulate a group Steiner Tree problem instance to determine the nodes to visit in the CTP tour [**?**] using the approach

in [59] coupled with the technique introduced in [84]. Due to the relatively small number of drop box locations, we optimally solve the TSP over selected locations using the Gurobi 9.1 MIP solver.

Our computational studies suggest that that proposed heuristic method approximates the Pareto frontier between cost and the minimum access function value well and does so quickly. Figure **??** plots the cost and minimum access function of solutions found using the MIP solver (**??**) against heuristic solutions (**??**) for four instances of the Milwaukee case study, each corresponding to covering sets defined by a different factor (1.0, 1.1, 1.2, or 1.3). We also plot the cost and minimum access function value of the rules-of-thumb (**??**) discussed in Section 4.5.1; however, recall that these policies may not be feasible for the DBLP. We find two primary benefits of the heuristic method. First, a large number of solutions are identified quickly. Our experiments showed that the MIP solver can find a new policy every 225 seconds on average when using lazy constraints, while the heuristic method is able to identify between 141-181 policies in 126-220 seconds in total, depending on the factor used to construct the covering sets. Second, the difference between the cost and minimum access function value of solutions identified by the heuristic and MIP methods is small. Moreover, the heuristic is able to identify solutions that are feasible for the DBLP that have a lower cost and higher minimum access function value than rules-of-thumb which may not be feasible for the DBLP. Election officials can implement the heuristic solutions or use them to determine a range of appropriate r values and then explore optimal solutions within this range.

We also investigate the heuristic on randomly generated DBLP instances to further assess its performance. The random instances were generated as follows. The location of voter populations and drop boxes were randomly selected within a 100 by 100 grid. We let the time to travel between each location be the Manhattan ($\ell 1$) distance. We randomly select up to a quarter of the drop boxes be required locations (T). Fixed costs were randomly



(a) Factor of 1.0

(b) Factor 1.1

(c) Factor 1.2

(d) Factor 1.3

Figure 4.6: Approximate Pareto frontier identified by the heuristic method (——) compared to MIP solutions (—△—). Blue dots (●) represent the cost and minimum access function value of the rules-of-thumb discussed in Section 4.5.1, with blue stars (⋆) indicating feasible solutions for $q = 2$.

selected between \$5,000 and \$12,000 for each drop box. Operational costs were computed using the same assumptions described in Section 4.5, except costs were scaled by a random value between 0.5 and 1.5. The time threshold used to construct the covering sets were randomly selected between 15 and 50. If a larger threshold was required to ensure each covering set included at least two locations, we select the smallest threshold possible. The value of v_w^1 was randomly selected between 50 and 95 and $v_w^0 = 100 - v_w^1$ for each $w \in W$. The value of a_{nw} was computed using the formula $e^{2.5-d_{nw}/30}$ where d_{nw} represents the Manhattan distance between the w and n for each $w \in W$ and $n \in N$.

A comparison of solutions generated by the heuristic and MIP solver for nine randomly generated instances is provided in Table 4.5. For each DBLP instance, we use the heuristic to obtain an approximation set for the DBLP. For each heuristic solution in the approximation set, we solve the DBLP to optimality using a r equal to the minimum access function value admitted by the solution. We then compute the cost deviation of the solutions using the average percent difference in the cost of heuristic and optimal solutions corresponding to the same value of r. The results presented in Table 4.5 are consistent with the findings from the Milwaukee case study. The heuristic method requires substantially less time than the MIP solver to find solutions corresponding to the same r values. Moreover, the heuristic method finds solutions with small cost deviations, which indicates that the heuristic finds policies that are similar in cost and access to the optimal solutions.

Table 4.5: Comparison of the heuristic and MIP implementations for random DBLP instances. The cost deviation is calculated as the average percent difference in total cost for solutions corresponding to the same r. Each MIP instance was terminated after 3600 seconds if no optimal was found.

$\|W\|$	$\|N\|$	$\|T\|$	Number of Solutions	Heuristic Time (s)	MIP* Time (s)	Cost Deviation
100	50	8	59	5.22	23	0.52%
500	50	11	61	14.42	168	0.46%
1000	50	7	53	28.05	90	1.98%
100	75	4	122	35.54	3,939	2.76%
500	75	18	109	45.91	960	2.90%
1000	75	15	93	65.33	661	0.97%
100	100	9	162	84.11	23,644	4.33%
500	100	1	193	173.95	349,400**	8.01%**
1000	100	12	117	119.53	30,109	4.12%

* Ignoring duplicate optimal solutions.
** 108 instances terminated after 3600 seconds.

4.6 Conclusion

In this chapter, we have introduced a structured and transparent approach to support the planning of ballot drop box voting systems. We do so by formalizing the drop box location problem (DBLP) to identify a set of optimal drop box locations. The locations are selected to minimize cost while ensuring voters have access to the drop box system and drop box risks are mitigated. Using a real-world case study, we demonstrate that the DBLP identifies drop box locations that consistently outperform rules-of-thumb across multiple criteria. We also find that the trade-off between criteria is non-trivial and requires careful consideration.

Our research suggests that optimization is an important tool for designing the drop box infrastructure. Simple guidance for designing drop box systems, such as locating one drop box per 15,000 registered voters, or other rules-of-thumb may be overly-simplistic and can cause election administrators to overlook cost-effective drop box locations that address inequalities within the voting infrastructure. Strategic drop box locations can reduce the "cost" of voting to address inequity within the voting system while ensuring that all voters have equitable access to the drop box system. Future research can utilize the DBLP to answer additional drop box policy questions and support the drafting of legislation surrounding the use of drop boxes.

We introduce a lazy constraint approach to solve the DBLP to optimality. Computational experiments show that a single optimal solution to the DBLP can be found relatively quickly using this approach within a state-of-the-art MIP solver for moderately sized problem instances. However, the multiple goals of the DBLP often requires multiple solutions to the DBLP. We find that this can cause computational times to increase to levels that are unreasonable for practice. This motivates the need for exceptionally quick solution methods for the DBLP. We introduce a heuristic for the DBLP, and we demonstrate that the

heuristic identifies quality solutions quickly. Initial attempts at reducing the computational time needed to identify optimal solutions using cutting planes originally introduced for the CTP [78] proved unfruitful. Future research into the theory of the DBLP is needed to reduce solution times for exact methods.

The DBLP is intended to be a component of a larger suite of tools for supporting election administrators understand, assess, and ultimately design different facets of the voting infrastructure. Ideally, the DBLP and other operations research tools will eventually be integrated into an online platform designed to support election administrators in all aspect of elections planning. However, current research is limited. It largely overlooks the vote-by-mail system and the risks of voting systems. There is a substantial opportunity for the operations research community to support election planning by appropriately modeling voting systems and voting infrastructure. Future research is needed to understand the temporal aspects of risk, particularly in the absentee voting process, and determine best practices for mitigating against malicious and non-malicious attacks. The DBLP and future models can then be incorporated into a comprehensive tool to support election officials in designing the election infrastructure in a way that increases voter turnout. A key challenge within this space is the need to understand and incorporate models that describe how voters freely select from multiple voting pathways once the infrastructure is set. Voter choices ultimately determine the cost-effectiveness and performance of the voting system.

Chapter 5

Recommendations for self-assigning volunteers

5.1 Introduction

Spontaneous volunteers are individuals, trained and untrained, who are not affiliated with a formal response effort but who self-deploy and contribute their resources when a disaster occurs [4, 199]. Local spontaneous volunteers are often the first "first responders" due to their ability to quickly respond to incidents [4, 142, 199]. Numerous examples highlight the importance of local, spontaneous volunteers during disaster relief efforts [58, 141, 142, 199]. After the 1995 Hanshin-Awaji earthquake, an estimated 67% of the rescues were conducted by spontaneous volunteers [136]. After the 2015 earthquake in Kathmandu, local residents are credited as the true first responders and led to the rescue of many individuals [54]. After Hurricane Harvey which made landfall primarily in Texas in 2016, spontaneous volunteers formed the emergent "Cajun Navy"using social media platforms to coordinate aid to affected areas [3].

Despite the importance of spontaneous volunteers [199, 213], they have rarely been

considered in formal disaster response planning [174, 199]. Often, spontaneous volunteers are refused by formal relief organizations as they are unprepared to use their assistance and volunteers are often unprepared to provide necessary support [45, 199, 213]. This is due to the significant coordination, integration, communication, logistical, health, and safety challenges they present [199]. For example, consider three challenges related to legal liability of deploying spontaneous volunteers, maintaining privacy of actors within the system, and managing the contributions of spontaneous volunteers:

- Legal: any organization that assigns spontaneous volunteers to tasks may have a legal liability for cases of disruption to organized response efforts or morbidity to volunteers, victims, and emergency responders [11, 69, 199].

- Privacy: private information of spontaneous volunteers should be secure against release to the public. Moreover, private information of the individuals needing help should be protected from volunteers to the furthest extent possible without impacting relief efforts.

- Managerial: routing information to and predicting contributions from spontaneous volunteers requires understanding of complex improvisation by spontaneous volunteers.

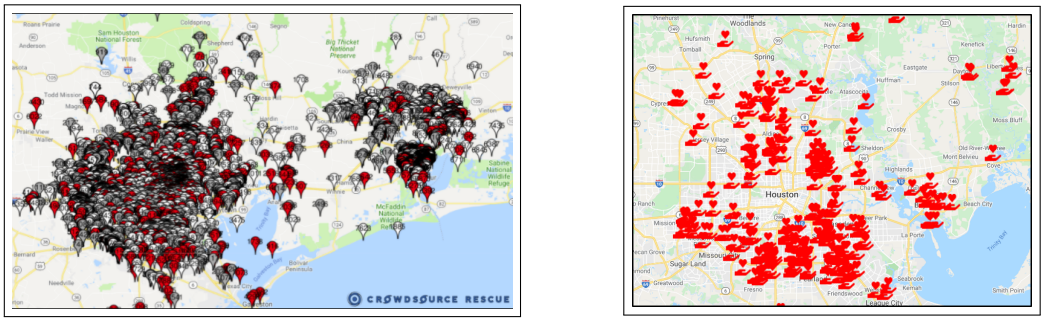
When spontaneous volunteers can be appropriately assigned to tasks matching their skills, substantial benefits to response efforts can be gained [213]. When turned away, either an immaterial cost in lost resource capacity is incurred [213] or spontaneous volunteers self-form "emergent groups" that improvise during the time of disaster [199, 213]. These emergent groups have a decentralized structure that allows them to be flexible during uncertain disaster response efforts [12]. However, the undirected, self-assigning nature of emergent groups is inefficient when providing aid due to a lack of coordination between

actors [12, 69, 102, 114, 199]. This leads to over service of some tasks and under-service of others [12].

In this chapter, we consider the role of spontaneous volunteers in completing "low acuity" tasks—those in which an immediate response from high skilled individuals is not needed. High acuity tasks that require a highly skilled team should be completed by trained personnel, such as urban search and rescue teams. However, the ability to effectively predict and manage spontaneous volunteer contributions can free up capacity of highly trained personnel, allowing them to respond to the complex cases for which they are trained [136, 154, 213] and reduce the need for additional trained search and rescue teams [69].

Fortunately, there is a growing number of organizations trying to plan for and employ spontaneous volunteers in some form [46, 12]. Recently, social media and online platforms have served as catalysts for spontaneous volunteers to coordinate ad hoc relief efforts and to do so at scale. These resources allow individuals to share and obtain information otherwise inaccessible to them [49, 102, 106]. More formal and specialized platforms [46] have started to emerge, but, like social platforms, they lack efficient allocation of volunteer resources to disaster relief tasks.

The motivation for the research in this chapter originates from CrowdSource Rescue [46], a non-profit that has created an online crowdsourcing platform to integrate and coordinate spontaneous volunteers into disaster relief efforts. The online platform serves as an information routing system to facilitate matchings between volunteers and tasks. Vetted spontaneous volunteers visit the platform to view and select from user-submitted tasks. While a powerful tool, the platform is prone to search friction, task starvation, and over-service of tasks due to the decentralized nature of how volunteers select tasks on the platform. Figure 5.1 illustrates a snapshot of the user interface during two disaster relief efforts. Such a user interface makes it difficult for users to identify which tasks



(a) (b)

Figure 5.1: User interface for spontaneous volunteers visiting the CrowdSource Rescue platform.

are important to complete and which tasks they prefer to complete. To improve the user interface and disaster relief efforts, we formalize the Online Task Recommendation (OTR) problem to construct personalized home screen "recommendations" that are displayed as default instead of or alongside the map of all tasks. These recommendations include fewer tasks to reduce search friction and help identify the tasks most likely to be of high preference to the volunteer but also societally important to complete (e.g., time sensitive and helps many individuals).

The OTR problem is difficult to solve exactly due to limited information during disaster operations, the interaction between independent volunteer choices, and the need to generate recommendation in real-time. In the presence of limited information, heuristic methods can be used to construct recommendations quickly. However, in many cases simple heuristics can perform poorly. We discuss one such case, illustrated in Figure 5.2, in which volunteers are recommended the tasks which they prefer the most, as is done with traditional recommendation systems. In this example, we consider a system with two volunteers and three tasks, and each volunteer is to be recommended two tasks. Suppose each volunteer shares the same preferences for the tasks. Quantitatively, we define the preferences as 1 for task 1, 0.2 for task 2, and 0.18 for task 3. Suppose that the volunteers select tasks randomly, but proportionally to the utility of other tasks recommended to them.

Using this approach, both volunteers are recommended tasks 1 and 2. Assuming each task only requires one volunteer and each volunteer completes one task, the expected number of tasks completed is 1.28. However, this can be improved by taking a more strategic approach. Suppose instead that volunteer A is recommended tasks 1 and 2 while volunteer B is recommended tasks 2 and 3. Under these recommendations, the expected number of completed tasks is 1.91, assuming both volunteers complete a task. Moreover, as long as the probability volunteer B completes a task remains above 0.31 (assuming volunteer A completes a task) when given the second recommendations, the second recommendations result in more completed tasks. This motivates the optimization based-approach presented in this chapter.

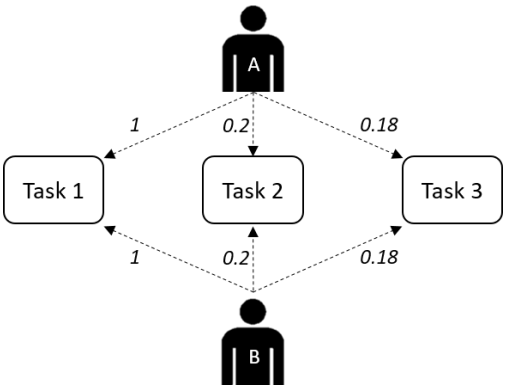


Figure 5.2: Example with two volunteers (A and B) and three tasks (1, 2, and 3). Arrows are labeled with the volunteer's preference for a task.

We present an integer optimization formulation based on matching models to automate the construction of personalized task recommendations for spontaneous volunteers. The formulation makes use of several guiding principles that reduces the amount of information required to generate useful recommendation. The optimization model selects and ranks up to a fixed number of tasks for each volunteer based on the societal importance of the task and the utility the volunteer gets from completing the task. We then introduce a

algorithm that, when a model of how volunteers select tasks from recommendations is known, improves the recommendations by changing the objective coefficients to avoid recommending a task that is likely to be completed by another volunteer.

As a proof of concept for this approach, we apply the methods to two case studies. First, we study the deployment of spontaneous volunteers during disaster response efforts following Hurricane Dorian using a Monte Carlo simulation. We then investigate how the proposed methods scale with larger, synthetically created data sets. In both case studies, we compare the proposed methods against recommending the k-closest tasks and a partially omniscient model that can act as a central controller and assign volunteers to tasks. We find the proposed methods outperform the closest assignment heuristic with respect to all considered metrics. We also find that the proposed optimization models perform relatively well compared to the partially omniscient central controller.

In Section 5.2, we present a literature review of relevant publications. In Section 5.3, we formalize the Online Task Recommendation Problem. In Section 5.4, we describe our optimization methods to generate recommendations for the Online Task Recommendation Problem in detail. In Section 5.5, we investigate the performance of the models using the two case studies. We conclude the chapter in Section 5.6 by discussing future research directions.

5.2 Literature review

The effective management of spontaneous volunteers during disaster response is a complex topic that requires research to be compiled from several fields, including disaster operations management, volunteerism, crowd sourcing, recommendation systems, and optimal matchings.

Table 5.1 summarizes the contribution of this chapter by comparing it to the closest

papers in the literature, which are presented in more detail throughout the literature review. We compare the contributions with respect to five categories. The first is whether the techniques presented in the paper are tailored for the management of spontaneous volunteers. As discussed later, this requires special consideration over traditional labor assignment problems. The second is whether the coordination of actors is largely computer mediated. Computer mediation allows for rapid communication and response by spontaneous volunteers. The third is whether task assignments or recommendations presented to the actors within the system are tailored to the actors' preferences or skill sets. As discussed later, a key component of volunteer retainment is to ensure volunteers are matched to preferred tasks. The fourth is whether actors within the system have discretion to take actions as they see fit, rather than assuming a central controller to control actors within the system. The fifth is whether actor contributions are directed through ranking of potential actions. This allows for free choice while also promoting societally important tasks. As seen in Table 5.1, there is no known existing literature to address or consider all five components. The contribution of this chapter is to introduce and study methods to rank potential tasks for spontaneous volunteers to promote public good but still allow for volunteers to retain flexibility in their response to disasters. This is notably absent from the existing literature.

5.2.1 Disaster Operations Management

Disaster operations management (DOM) is the management of activities related to disaster mitigation, preparedness, response, and recovery with the goal of reducing personal and economic impact so the impacted area can return to normalcy quickly following a disaster [43]. Literature on DOM covers a wide array of techniques and applications from the intermodal transport of food grain [115] to simulating the anthrax spore dispersion in a bioterrorism incident [156]. Much of the literature investigates and describes good policies for DOM, and, more pertinently, there is a plethora of management science and operations

Table 5.1: Comparison of Existing Literature

Paper	Spontaneous Volunteer Management	Computer Mediated	Tailored Tasks	Volunteer Choice	Ranked Recommendations
Abualkhair et al. [4]	✓				
Auferbauer et al. [12]	✓	✓	✓	✓	
Chaptini and Lerman [32]		✓	✓	✓	
Danaf et al. [50]		✓	✓	✓	
Falasca and Zobel [67]			✓		
Jiang et al. [97]		✓	✓	✓	
Lassiter et al. [110]			✓		
Lo et al. [113]	✓	✓	✓	✓	
Ludwig et al. [114]	✓		✓	✓	
Manshadi and Rodilitz [116]	✓	✓	✓	✓	
Mofidi and Pazour [133]		✓	✓	✓	
Mayorga et al. [121]	✓				
Paciarotti et al. [145]	✓	✓			
Paret et al. [148]	✓				
This chapter	✓	✓	✓	✓	✓

research (MS/OR) literature on the topic [8, 27, 62, 75, 93, 174].

Crisis informatics is a subfield that investigates the social, informational, and technical components to disaster preparation, response, and recovery. Primarily, it focuses on computer-mediated public participation in disasters [146]. Palen and Lin [146] investigate the communication channels of public participation during disasters. They argue that the command-and-control model used in disaster management by many formal organizations is outdated, especially considering the role of the public in disaster preparation and response and argue the need for new information and communication technologies to support public participation. Reuter and Kaufhould [157] conduct a survey of social media in emergencies and outline future directions in crisis informatics. They identify a need for a structured methodology to match needs and offers by spontaneous volunteers. Crisis informatics research has not focused on prescriptive methods to route information between actors within the system. This is a gap that our research begins to fill.

5.2.2 Volunteer Convergence

Volunteer labor assignment is decidedly different than traditional labor assignment and constructs guidelines for volunteer labor assignment [162]. Sampson [162] describes the need to ensure utilization of volunteer resources and to assign them high quality tasks to encourage reoccurring participation of volunteers. Fernandez [69] suggests that assigning volunteers to high quality tasks is non-trivial and argues the need for a volunteer management system. Sampson [162] investigates encoding the volunteer labor assignment guidelines into an integer program to assign workers to review conference papers. Gordon and Erkut [82] present a spreadsheet based decision-support tool for scheduling volunteers for a music festival. Kaspari [101] present a goal programming method for scheduling volunteers within a bike sharing program. In each of these papers, preference of volunteers is key factor, driven by the belief that giving volunteers tasks of high preference improves likelihood they take action and continue to volunteer in the future [162].

Some research has focused specifically on volunteerism within disaster management operations. Simo and Bies [173] find that spontaneous volunteer groups are critical to disaster relief efforts although there are cases where such resources are not manageable due to an overwhelming influx of resources. Simo [172] finds that using response frameworks at the time of research, it was difficult to cope with the large surge of volunteers. Oloruntoba [144] finds that volunteers play a large role in response and relief efforts, but volunteers can become an obstacle rather than a benefit without strategic planning. Fernandez [69] investigates spontaneous volunteers specifically and finds that a unique disaster management system is needed to effectively use spontaneous volunteer resources and ensure they don't hinder relief efforts. Paciarotti et al. [145] review the spontaneous volunteer management practices, specifically regarding those used during flooding in the Senigallia region of Italy. They identify strengths and weaknesses of the analyzed manage-

ment practices using surveys and interviews. Ludwig et al. [114] investigate coordinating the efforts of spontaneous volunteers through the use of public displays. Auferbauer et al. [12] presents a workflow design and field experiments for a crowdtasking prototype application in crisis and disaster management to directly assign volunteers to tasks. The framework requires volunteer information to be submitted via an application and once a task requiring a worker matching their description is submitted, an alert is sent to the volunteer. The volunteer can then accept or reject the task [12]. Manshadi and Rodilitz investigate the frequency at which such alerts should be sent to each volunteer and identify the opportunity to investigate how to alert users of a subset of tasks, rather than a single task, while considering individualized discrete choice functions as a future research direction [116].

Six known papers address volunteer and manpower planning for disaster operations management. Falasca and Zobel [67] propose a biobjective optimization model to assist in the assignment of volunteer resources in humanitarian organizations using a series of principles from the field of volunteer management. Lassiter et al. [110] introduce a bi-objective robust optimization framework for assigning volunteers to maximize volunteer assignment utility and minimize unmet demand. Mayorga et al. [121] investigate the assignment of spontaneous volunteers with stochastic server arrival and abandonment using a multi-server queuing system. Abualkhair et al. [4] use queuing theory to investigate how to plan response efforts considering the stochastic nature of (i) beneficiary arrival and service times, (ii) relief donation arrival and service times, and (iii) volunteer arrival and abandonment times at disaster relief sites. Lo et al. [113] investigate optimal policies with respect to volunteer commitments on a matching platforms to address food shortage problems within the United States while maintaining high levels of volunteer engagement. Paret et al. [148] investigate optimal assignment and heuristics policies for volunteers during disaster scenarios with stochastic arrival of demand and volunteers using Markov decision

processes. No known operations research literature investigates allocating spontaneous volunteer resources without an organization which assigns tasks.

5.2.3 Crowd Sourcing

Crowd sourcing systems are defined by workers, tasks, and a platform. Tasks are submitted to a platform and workers are either assigned to or select tasks from this platform. Chittilappilly et al. [35] survey the available general crowd sourcing techniques. These systems are defined by, among many other things, the way tasks are assigned to workers. There are two prevalent platform designs today with resepect to task assignment: worker selected tasking (WST) and server assigned tasking (SAT) [103]. In a WST system, workers select from the active tasks published to a public database. In WST systems, each individual worker selects tasks based on their own preference which may not align with the preference of the the platform [103]. Moreover, in WST systems self assignment is known to lead to an inefficient allocation of resources to tasks, known as starvation [108], and so-called search friction issues from which workers have difficulty finding appropriate tasks [89]. In a SAT system, each worker is assigned a task(s) to complete from all of the active tasks by the platform, a central controller [103]. In a SAT system, some platform designs compromise the privacy of the worker as the location of every person within the system must be known [196].

With general crowd sourced tasking, task assignment is not considered to be one of the largest issues [35]; however, in spatial crowd sourcing, task assignment is the key issue [196]. Spatial crowd sourcing focuses on assigning workers to tasks but accounts for the location of the worker and the task. Kazemi and Shahabi [103] present a taxonomy for spatial crowd sourcing. They divide systems by incentives (reward-based or self-incentivising), publishing methods (WST or SAT), amount of information known apriori (offline or online) and redundancy (one or many workers per task). There is limited

research with regards to WST systems for task assignment with Deng et al. [52] providing the only readily available literature. They investigate what route workers should take to complete tasks once the workers select the tasks they wish to complete [52]. SAT systems are much more widely investigated for task assignment, and most rely on the use of optimization tools. Approaches for SAT system assume that workers can be directly assigned a single tasks which they may choose to accept or not accept [89, 194, 34, 200]. Some approaches address privacy [194], the multi-period nature of assignments [195], or problem trade-offs [200]. Only one known paper investigates a WST and SAT hybrid design where the platform strategically reduces the number of options from which the workers choose but avoids direct assignment of tasks to workers. Mofidi and Pazour [133] study a bi-level integer programming approach to recommend a collection of tasks to freelance suppliers in a way that improves the amount of benefit the platform receives from completed tasks. Their work differs from that studied in this chapter, because they assume suppliers always select the highest utility recommended task and task utility for each supplier is known exactly.

5.2.4 Recommendation Systems

Recommendation systems condense a large number of options to highlight and recommend what is believed to be high utility options for each user of a system [159, 208]. Traditional recommendation systems use one or more of three approaches: collaborative filtering, content-based filtering, or demographic-based filtering. Collaborative filtering predicts user preference by identifying users who like similar items and for who the preference for a new item is known [139]. Content-based filtering predicts user preference by comparing the features of an item to those of items with known preference [139]. Demographic-based filtering predicts user preference based on the preference of users with similar characteristics [139]. In general, recommendation systems attempt to estimate user preference for

items and recommend the most preferred items.

Recommendations using discrete choice modeling first introduced by Chaptini and Lerman [32] are related to the topics discussed in this chapter. Discrete choice modeling defines the likelihood an individual selects an alternative from a set of provided options. Common discrete choice models include probit, logit, and multinomial logit models. These models assume rationality of user choice under the assumption that the true preferences are unknown but follow specified distributions [7]. Chaptini and Lerman [32] define a procedure to estimate user preference of items by modeling users interaction with a system with a discrete choice function. Recommendations are then generated by recommending the items with the highest predicted utility or preference. Jiang et al. [97] use discrete choice models to measure the probability that a user will choose at least one item from their recommendations to increase option diversity and relevancy simultaneously. They formulate the problem using a nonlinear integer program. Danaf et al. [50] investigate a Hierarchical Bayes approach to estimate and update user preferences for an app-based recommendation system providing a personalized menu of options to users. In general, there are few application of discrete choice modeling in recommendation systems. In a similar vein is the field of assortment optimization. Assortment optimization consists of selecting a subset of items to offer to customers. They consider the added goal to maximize revenue, potentially with budget or cardinality constraints [107]. The customer is often assumed to purchase items with a probability described by a discrete choice model with a "no purchase" option that becomes more likely if the offered goods are undesirable [107]. However, assortment optimization does not aim to offer customers with personalized product offering [107], and we could not find model variations that address a problem similar to the one addressed in this chapter.

We find that existing techniques in the recommendation system literature are insufficient for the application considered within this chapter. First, the goal of recommendation

systems in the literature is to recommend each individual their most preferred tasks and the challenge is to learn the preferences. In this chapter, the construction of recommendations must also consider the impact the recommendations and subsequent choice from those recommendations on the overall system. For example, if a volunteer selects an item from their recommendations, then no other volunteer can complete the task. In this way, the tasks are scare goods that must be effectively distributed to volunteers. Second, existing methods are founded on discrete choice models that assume users are rational and given a set of recommendations, they will always select the option that provides them the highest utility. However, order effects are a well studied phenomenon where the order in which options are presented to a user impacts the likelihood of selecting an option [117]. Order effects generally manifest as *primacy* or *recency* [117]. Moreover, Joachims et al. [98] show that user choice is impacted by the suggested ranking from a recommendation system based on Internet search results. Order effects are likely to have a large impact on the tasks which volunteers select. It has been noted by disaster relief experts that volunteers tend to complete the first tasks when given a set of options [11]. It is likely that volunteers wish to complete the highest ranked tasks as they believe they are the most impactful and important tasks. Existing methods are insufficient to address the impact of order effects in a recommendation system. Lastly, the proposed discrete choice-based recommendation techniques assume that discrete choice models can be learned for each volunteer. This may be infeasible due to short-lived nature of disaster relief efforts and the presence of unique tasks that are completed only once.

5.2.5 Optimal Matchings

Maximal weight matchings are a well studied problem which aim to match elements of a set in a way that maximized the value of all pairings. The bipartite matching problem focuses on a matching problem with two disjoint sets and elements of one set can only be

matched to elements of the other [42].

Variants to traditional bipartite matchings have also been investigated. The capacitated house allocation problem aims to assign residents to housing when the residents have preference on their housing [175]. The hospitals/residents problem aims to match many residents to multiple hospitals when both residents and hospitals have preference [96]. It is common in these problems to ensure a "popular matching"—one in which no other matching is preferred by more elements– rather than maximizing the total value of the matchings [175]. These problems consider a many–to–one matching where multiple items of one set can be assigned to one element of the other. There is limited investigation into many–to–many matching problems. Colannino [39] presents a many–to–many matching formulation that requires each element from two disjoint set be matched with at least one element of the other. The cost of this matching is then minimized.

Online matching problems are related to bipartite matching problems and add a temporal aspect. Online matching problems are defined by a set of servers, known apriori, which must be matched with requests that dynamically arrive over time [100]. The benefit gained from matching each server to each request is not known a priori and becomes known when the request is revealed [100]. Once a server is matched to a request, it cannot be matched to a request that arrives later [100]. The goal is to match servers to requests as to maximize the value of the matchings [100]. There are many extensions of this problem, including a penalty to delay matchings and the ability to change a limited number of past matchings once more requests are revealed.

Matching platforms use matching methods within platforms, such as dating or job recruiting platforms, where preference is two-sided and the goal is to maximize the value of selections from matches. Over time, they learn user preferences to provide quality matches to elements of both disjoint sets with the hopes that matched elements will eventually both "select" each other. A common research goal when investigating matching platforms is

determine and investigate optimal subscription fees [10]. Halaburda et al. [87] investigate a two-sided preference matching platforms and consider a two-sided matching problem in which elements of two disjoint sets must select each other after being matched. Among many things, they find that matching platforms that strategically reduce the options presented to actors within a system can provide more value than platforms that present every option to all users, under the right conditions.

The matching problem presented in this chapter differs from formulations available in the literature. We consider a many-to-many maximal weighted matching problem, where an element of one disjoint set can be matched to no more than a fixed number of elements from the other set. We assume a one-sided preference model. Our problem also ranks the matchings for elements of one of the disjoint sets with one-sided preference. No readily available literature investigates this problem.

5.3 Online Task Recommendation Problem

We formalize the problem of generating personalized recommendations for a default home screen as the Online Task Recommendation (OTR) problem. We consider an online WST platform to which tasks and associated information (e.g., location, worker requirements) are submitted and routed to workers. In this context, each worker within the system is a spontaneous volunteer (or a group of volunteers).

The disaster relief effort is divided into discrete, mutually exclusive time epochs as described by Figure 5.3. These discrete epochs can be heterogeneous in length and could be defined by changes to the system (e.g., a task is completed or a task enters the system). The epoch length should be of a sufficient length to allow for data cleaning and to limit frequent changes to volunteers recommendations. Each epoch is comprised of two phases: the Response Phase and the Construction Phase. Each of the sets and parameters introduced

in the subsequent paragraphs may change over time. For simplicity, we do not include a subscript t indicating the value in epoch t.

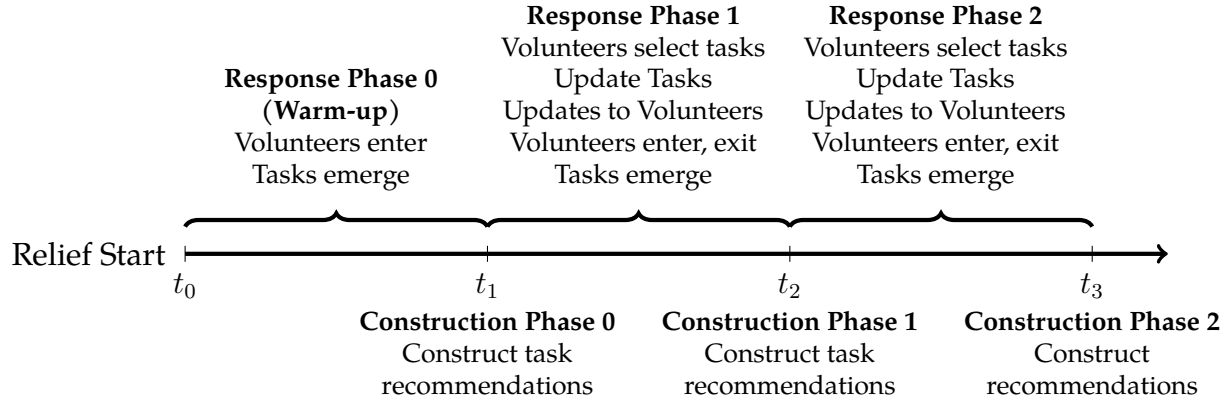


Figure 5.3: The discrete epoch timeline of event for which our methods are designed. After a warm-up, there is a Construction Phase in which recommendations are generated and given to volunteers based on the current state of the system. After the recommendations are constructed, the volunteers have the option to choose from the recommendations and respond in the Response Phase. Two-phase process is repeated until relief efforts are completed.

Let D be the set of tasks whose elements need to be *completed* by exactly one volunteer from the set of volunteers, S. We let each task $d \in D$ have a societal value v_d of being completed (i.e., societal importance of completing task). Societal value could be measured in many ways but could, for example, consider the number of individuals the task aids, the certainty of the associated task data, the amount of time that has passed since the task was added to the platform, and the dependence of other tasks on the completion of the task of interest. Each task $d \in D$ *emerges* or is submitted to the platform at different times during the disaster relief effort. Likewise, each volunteer $s \in S$ can *enter* and *exit* the disaster relief effort. Each volunteer $s \in S$ has utility u_{sd} of completing each task $d \in D$. The utility represents volunteer s's perceived benefit of completing task d and may not align with the societal value, v_d, of the task being completed. We assume an estimate of the utilities u_{sd} or a function to calculate the estimated utilities is known. A utility estimate may be

made by comparing the needs of the task and a *preference profile* for a volunteer, which considers their current location, skills, type of vehicle, and preferences of who to help (pets, medically disabled, children/adults/elderly). Estimating these utilities during short-lived disaster relief efforts is a nontrivial task, but existing research on eliciting utility functions [212] or utility estimation techniques from traditional recommendation systems may be applicable. Moreover, with careful consideration of the exploration-exploitation trade-off, these user utilities may be able to be learned over time based on volunteer interaction with the system. Throughout this manuscript, we assume that utility estimates are made prior to generating recommendations.

In the initial Response Phase, or Warm-up, tasks emerge and volunteers enter the system. The system remains in this stage until a sufficient number of tasks emerge and volunteers enter to construct recommendations. Once these conditions are met, the first Construction Phase occurs.

In the Construction Phase, a subset of up to L active tasks are recommended to each active volunteer in the form of an ordered list. The L tasks comprise the list of recommendations presented on the default homepage of an application. More than L is assumed to result in search friction issues. Fewer then L could reduce the likelihood that the volunteer participates in the current or subsequent time periods or even lead to legal liability issues. The tasks on a list must be unique (i.e., we cannot recommend a task more than once to the same volunteer in the same epoch). It is desired that L tasks are recommend to each volunteer unless fewer than L tasks provide positive utility to the volunteer or other constraints restrict the recommendation of more tasks. In some cases, the tasks that can be recommended to a volunteer may be restricted. The sets S_d and D_s describe the volunteers that can be recommended task $d \in D$ and the tasks that volunteer $s \in S$ can be recommended, respectively. Note that, for a given s and d, $d \in D_s$ if and only if $s \in S_d$. The use of the sets S_d and D_s importance is three-fold. First, there may be natural desire to

avoid recommending certain tasks to certain volunteers; for example, infrastructure may be compromised and it could be too dangerous to recommend tasks in some impacted areas. Second, for large problem instances, these sets can be used to limit the number of variables considered in a mathematical formulation to reduce solution times and computational requirements. Third, these sets could be used to enhance volunteer privacy. Volunteers could provide district-based locations rather than exact locations, and then the tasks recommended to them could be required to be in the same or neighboring districts. This similar to an approach used by To et al. [194] when examining privacy within crowd sourcing platforms.

In the Response Phase, each volunteer $s \in S$ selects and *responds* to a task $d \in D_s$. We do not assume a discrete choice model to describe how volunteers select from their recommendations is known due to the short lived nature of the disaster relief effort. In Section 5.4.3 we address the case when such a function is known. Often, the volunteer responds to a task from their recommendations, but it is possible a volunteer responds to a task not recommended to them or does not respond to any task during the epoch. If more than one volunteers respond to a task in the response phase, only one can *complete* the task and the value of the task is *captured* only once. When more than one volunteer responds to a task, the task is said to be *over served* and the response is inefficient. If more than one volunteer responds to a task, then the volunteer with the highest utility for the task completes the task. This one utility is *captured* during the relief efforts. The volunteers that do not complete the task get a utility of zero from responding to the task. Once a task is completed, the platform is informed and the set of active tasks updated. It is possible that some tasks are not completed during a response phase. These tasks will either remain to be completed within the next epoch or abandon the system (e.g., completed by a formal relief organization).

Within this framework, the OTR problem is to determine which tasks to recommend to

each volunteer and what ranking or position to give each recommended task during each Construction Phase. Depending on the situation, there may be multiple objectives for the OTR problem. For example, objectives may be to maximize the captured utility, maximize the value of completed tasks, minimize the time to complete the tasks, or minimize the over service of tasks.

5.4 Model Formulation

The OTR problem is challenging for a number of reasons. First, we do not assume a function to predict how volunteers select from their recommendations is known, and therefore, we cannot predict which of the active tasks will be active in future epochs. Furthermore, tasks can be completed by non-spontaneous volunteer actors (e.g., a formal relief organization) within the system whose contribution we cannot predict. Moreover, new tasks can be submitted to the platform and, currently, there are no technologies to accurately predict the location of or preference for these tasks. In addition, volunteers may enter the system and others may leave the system stochastically due to factors outside the disaster response effort. Significant advancements in empirical knowledge about disaster response efforts are needed to instantiate a model of the system in such detail. Without such advancements, predicting future states and the likelihood of them cannot be reasonably integrated into the generation of recommendations. As a result, we present the Maximal Ordered Multiple Matching (MOMM) problem as an optimization method to construct quality recommendations for the OTR problem quickly using limited information about the system. We then show that such an approach performs reasonably using case studies in Section 5.5.

The collection of sets, parameters, and variables used in the MMOM model are provided in Table 5.2. The MOMM model makes use of the following four guiding principles. First,

the MOMM is founded on the principle that recommending tasks with high v_d increases the likelihood that high value tasks are completed, and recommending tasks to a volunteer such that u_{sd} is large increases the likelihood a volunteer has reoccurring participation within the disaster relief effort [162]. Second, the MOMM model is founded on the principle that recommending a task $d \in D$ to volunteer $s \in S$ for which the product $v_d u_{sd}$ is large aligns with the OTR problem objectives. Pairings such that both v_d and u_{sd} are both large are desired as a high value task is recommended to a volunteer who has a high preference for the task. Pairings such that both v_d and u_{sd} are both small are not desired as a low value task should not be recommended to a volunteer who has a low preference for the task. Pairings such that either v_d or u_{sd} are large are moderately desired. Third, the MOMM model is founded on the principle that the probability that a volunteer selects a task depends on the ranking given to the task, and it is desirable to rank tasks with a large product $v_d u_{sd}$ in positions most likely to be selected by volunteers. We introduce a weight w_i associated with each position $i = 1, \ldots, L$ to capture this effect. Fourth, the MOMM model is founded on the principle that the over-service of highly preferred tasks and task starvation of highly-unpreferred tasks can be limited by reducing or increasing the number of volunteers to which it is recommended, respectively. We introduce constraints such that a task $d \in D$ may be recommended to at most J_d^u and no fewer than J_d^ℓ volunteers.

5.4.1 Maximal Ordered Multiple Matching 0,1-Integer Program

The MOMM model is formulated as a one-sided preference bipartite matching problem in which multiple elements of the set D can be matched with (i.e., recommended to) multiple elements of the set S. These matchings are simultaneously ordered (i.e., given a ranking). We let the decision variables be $x_{isd} = 1$ if task d is recommended to volunteer s at the i^{th} position, for $s \in S,\ d \in D,\ i = 1, \ldots, L$, and 0 otherwise.

The formulation of the MOMM model is presented below.

Table 5.2: Notation used in the MMOM model

Sets

S	set of volunteers within the system
D	set of tasks in the system
$D_s \subset D$	subset of tasks which volunteer $s \in S$ can be recommended
$S_d \subset S$	subset of volunteers which can be recommended task $d \in D$

Parameters

L	the maximal number of tasks recommended to each volunteer
J_d^ℓ	minimum number of recommendation lists on which a task $d \in D$ can appear
J_d^u	maximum number of recommendation lists on which a task $d \in D$ can appear
u_{sd}	the utility task $d \in D$ provides to volunteer $s \in S$
v_d	societal value of completing task $d \in D$
w_i	weight associated with the i^{th} entry on the ordered list, $i = 1, \ldots, L$ with $\sum_{i=1}^{L} w_i = 1$ and $w_i \geq 0$

Decision Variables

$$x_{isd} = \begin{cases} 1 & \text{if task } d \in D_s \text{ is placed at position } i \text{ on volunteer } s\text{'s list}, i = 1, \ldots, L,\ s \in S \\ 0 & \text{otherwise} \end{cases}$$

$$z^* = \max \sum_{i=1}^{L} \sum_{s \in S} \sum_{d \in D_s} w_i v_d u_{sd} x_{isd} \tag{5.1}$$

$$s.t. \sum_{i=1}^{L} \sum_{s \in S_d} x_{isd} \geq J_d^\ell \quad \forall\, d \in D \tag{5.2}$$

$$\sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} \leq J_d^u, \quad \forall\, d \in D \tag{5.3}$$

$$\sum_{d \in D_s} x_{isd} \leq 1, \quad \forall\, i = 1, \ldots, L,\ s \in S \tag{5.4}$$

$$\sum_{i=1}^{L} x_{isd} \leq 1, \quad \forall\, d \in D,\ s \in S_d \tag{5.5}$$

$$x_{isd} \in \{0, 1\}, \quad \forall\, i = 1, \ldots, L,\ s \in S,\ d \in D_s \tag{5.6}$$

The objective (5.1) is to maximize the sum of the product of the position weight, task value, and utility of completing the task, which describes the benefit of the matching. As a result, the ranking of the tasks is such that the task with the largest $v_d u_{sd}$ matches with the index i with largest weight w_i, second largest $v_d u_{sd}$ matches with the index i with second largest weight w_i, and so on. Constraint sets (5.2) and (5.3) ensure the number of lists each task $d \in D$ appears on is between J_d^ℓ and J_d^u. Constraint set (5.4) ensures that at most one task is recommended in each position on the recommendation list. Constraint set (5.5) ensures that the list is unique, or that no task appears on a volunteer's list twice. Constraint set (5.6) requires the variables to be binary.

Theoretical Properties

The constraint matrix, A, of the standard linear programming relaxation to the MOMM model in the form $Ax \leq b$ is totally unimodular, as presented in Theorem 5.1. The standard linear programming relaxation of the MOMM model has the same constraints, except that constraint set (5.6) is relaxed to the form $0 \leq x_{isd} \leq 1$.

Theorem 5.1. *Let A be the constraint matrix for the MOMM problem standard linear programming relaxation in the form $Ax \leq b$. The matrix A is totally unimodular (TU).*

The proof to Theorem 5.1 is provided in Appendix 7. Given integer values of J_d^ℓ and J_d^u for each $d \in D$, the right hand side vector, b, has integer entries. Thus, as A is totally unimodular, there exists an optimal solution to the standard linear programming relaxation with integer variable values, and the solution is an optimal solution to the original integer program. Thus, given integer J_d^ℓ and J_d^u values, the MOMM model can be solved using linear programming. As a result, the MOMM problem has a complexity at most that of linear programming, which is known to be polynomial in the number of variables [206].

The the number of variables (including slack and surplus variables) for the MOMM model is $\mathcal{O}(|S||D|L)$.

Additionally, as stated in Theorem 5.2, the MOMM model can be reformulated as a minimum cost network flow problem in polynomial time. The proof is provided in Appendix 7. The minimum cost network flow formulation may be useful as polynomial time algorithms for solving minimum cost network flow problems are known and could potentially reduce solution times for large problem instances. However, initial investigation of the standard implementation of the minimum cost network flow problem are slower to solve than when using linear programming to solve the standard linear programming relaxation of the MOMM problem.

Theorem 5.2. *Given $J_d^\ell \in \mathbb{Z}$ and $J_d^u \in \mathbb{Z}$ for $d \in D$, the MOMM model can be reduced to a minimum cost network flow problem in polynomial time.*

5.4.2 MOMM in a Multi-epoch Setting

The MOMM model creates recommendations for a single time epoch. In this subsection, we describe how to apply the MOMM model to a multi-epoch situation as described in the OTR problem. The general procedure follows the logic described in Figure 5.3.

The disaster relief effort begins with a warm-up epoch. When volunteers enter the system, they will identify themselves to the platform as available to complete tasks and specify preference for certain types of tasks. This can be done in practice by having volunteers complete a user profile. The end of the warm-up epoch initiates the start of the first construction phase.

The Construction Phase entails the following steps:

1. Estimate relevant parameters:

(a) Load the set S of the active volunteers and set D of active demands.

(b) Update the appropriate sets S_d and D_s for all $d \in D$ and $s \in S$.

(c) Recompute the appropriate values J_d^ℓ for each $d \in D$, J_d^u for each $d \in D$, and w_i for each $i = 1, \ldots, L$.

(d) For each task $d \in D$ re-estimate the value v_d.

(e) For each task $d \in D$ and each volunteer $s \in S_d$ re-estimate the utility u_{sd}.

(f) Compute the object function coefficients $u_{sd}v_dw_i$.

2. Build and solve the MOMM model, generating recommendations for all volunteers simultaneously.

3. Alert the volunteer of task recommendations and display the recommendations to the volunteer

Step 1 compiles and computes the appropriate parameters and sets. The sub-steps in Step 1 are introduced without specifics as each application of the MOMM model is likely to have unique considerations that require the computation of any of the parameters or sets to be changed. For example, the privacy settings of the platform's users and disaster-specific conditions may determine which tasks can be recommended to each volunteer $s \in S$ as defined by the D_s. The order in which sub-steps of Step 1 are completed can change as appropriate to the situation. In Step 2, we build the MMOM model using the parameters and sets computed in Step 1, and then solve the model. The solution of the model gives the recommendations, and we present the recommendations to the user. In Step 3 we alert the user of the recommendations. The frequency at which alerts are sent to volunteers is an interesting question itself that some research has begun to investigate [116].

The end of the Construction Phase initiates the start of the next epoch and the Response Phase. During this phase volunteers may enter or exit. When volunteers enter, we wish to

provide them with recommendations immediately. However, we may not want to continually change the recommendations for other volunteers. To construct recommendations for entering volunteers, a modified construction phase is run that "locks" the recommendations given to the existing volunteers (x_{isd} variables set equal to 0 or 1 according to existing recommendations). The other steps of the Construction phase remain the same. An area for future work is to determine the optimal frequency at which to update recommendations to avoid alert fatigue and promote reoccurring volunteer participation.

5.4.3 Objective Coefficient Modification Algorithm

The MOMM integer program does not explicitly take into consideration the interaction between volunteers when responding to tasks during the Response Phase of the OTR problem. This is because it is not assumed that a function to describe how users select tasks from their recommendations is known. In this section, we consider the case in which such a function is known for each volunteer, and we describe how to modify the MOMM model to improve recommendations. We introduce the objective coefficient modification (CM) algorithm to modify the objective function coefficients used in the MOMM model. The CM algorithm is a heuristic method founded on the principle that a task should not be ranked in positions with large weights w_i or recommended at all to volunteers if there is a high likelihood the task would be completed by another volunteer who prefers the task more.

To understand why the CM algorithm may be useful, consider the following example. Let there be three volunteers and three tasks. Let $L = 3$ and consider the case where we let $J_d^u = 3$ for $d \in D$ so that each volunteer receives three recommendations. Suppose two of the tasks provide little societal value or utility to any of the volunteers. The third, however, provides a large societal value and utility to all volunteers. Using the MOMM, the third task will be the top recommendation for each volunteer, and in many situations all three volunteers will respond to the same task, but only one volunteer is required. Instead,

consider the recommendations such that the third task remains the top recommendation for volunteer one but becomes the last recommendation for volunteers two and three. Assuming w_i is decreasing as i increases, the likelihood that task one or two is completed increases, and we expect the volunteers to collectively complete more tasks. The CM algorithm automates the process of reweighting tasks.

Let the function $F_s(d \mid r_s)$ return the probability that volunteer $s \in S$ selects and respond to task $d \in D$ given ranked recommendations r_s. The CM algorithm procedure iteratively generates recommendations using the MOMM model and then updates the objective coefficients for MOMM model by multiplying $w_i v_d u_{sd}$ by p_{sd}, the probability that a volunteer $s \in S$ would complete a task $d \in D$ (i.e., have the highest utility of all responding volunteers) if s responded to it, for all $i = 1, \ldots, L,\ d \in D,\ s \in S$. The value of p_{sd} is calculated using the function $F_s(d \mid r_s)$. The resulting product $p_{sd} w_i v_d u_{sd}$ is used as the objective coefficient to generate the recommendations in the next iteration. The algorithm terminates once the expected benefit does not increase. Expected benefit is calculated as the product of task value v_d and expected utility captured. The algorithm outputs the final coefficients $p_{sd} w_i v_d u_{sd}$ for all $i = 1, \ldots, L,\ s \in S,\ d \in D_s$ to use with the MOMM model to construct the recommendations. When the CM algorithm is used within the application of the MOMM model, Step 1(e) as described in Section 5.4.2 is modified. The modified step is to run the CM algorithm to determine the objective function coefficients.

The logic of the CM algorithm is:

0. Initialize:

   (a) Denote as $S_{sd} \subset S$ the volunteers that have a higher or equal utility of completing task d than s. Ties between volunteers with equal utility should be broken by some deterministic rule so volunteers with equal utility are not in each other's set S_{sd}. For example, each volunteer could be assigned an index, and, in the case

of equal utilities, volunteers with a lower index than s are included in S_{sd}.

(b) Objective coefficients for the MOMM model as $c^0_{isd} = w_i v_d u_{sd}$ for all $i = 1, \ldots, L,\ s \in S,\ d \in D_s$.

(c) $n = 0$.

1. Repeat until termination:

(a) Let $p^n_{sd} = 1$ for all $s \in S$ and $d \in D_s$.

(b) Solve the MOMM problem using objective coefficients c^n_{isd} for each variable x_{isd}.

(c) Let the ranked recommendations for each volunteer $s \in S$ be r^n_s.

(d) For each $s \in S$ and $d \in D$, set $p^n_{sd} = \Pi_{s' \in S_{sd}}(1 - F_{s'}(d \mid r^n_{s'}))$, the probability that none of the volunteers $s' \in S_{sd}$ respond to task d. Note this calculation assumes independence between the probability any two volunteers select a task given their recommendations, but could easily be adjusted to account for dependence.

(e) Set the objective coefficient $c^{n+1}_{isd} = p^n_{sd} w_i v_d u_{sd}$ for each $i = 1, \ldots, L,\ s \in S,\ d \in D$.

(f) Terminate if the expected benefit

$$\sum_{d \in D} v_d \sum_{s \in S_d} u_{sd} F_s(d \mid r^{n+1}_s)\ \Pi_{s' \in S_{sd}}(1 - F_{s'}(d \mid r^{n+1}_{s'}))$$

is not strictly larger than when using coefficients c^n_{isd} where r^{n+1}_s are the recommendations when using coefficients c^{n+1}_{isd}. Return coefficients c^n_{isd}.

(g) Else, $n = n + 1$ and return to step 1(a).

The CM algorithm terminates in a finite number of iterations, since there is a finite number of possible recommendations. The same set of recommendations results in the same expected benefit captured during any iteration. With each iteration, the new coefficients increase the expected benefit of the recommendations or the algorithm terminates. As a

result, the same solution never repeats without termination. In practice the CM algorithm terminates in relatively few (≤ 10) iterations.

The CM algorithm is also guaranteed to generate coefficients that lead to an expected benefit captured at least the same as using the standard coefficients, $w_i v_d u_{sd}$. Notice in the first iteration of the CM algorithm, the algorithm solves the MOMM model with the standard coefficients. With each iteration, the algorithm either find coefficients that lead to a higher expected benefit or terminates and returns the coefficients that provided the highest expected benefit.

5.5 Case Studies

We present two case studies to demonstrate the efficacy of the MOMM model and CM algorithm during disaster relief efforts. In Case Study 1, we simulate volunteer response efforts following the landfall of Hurricane Dorian on the Grand Bahamas Island, The Bahamas using data collected from the the CrowdSource Rescue Platform [46] and Monte Carlo simulation. In Case Study 2, we study the scalability of the proposed methods with randomly generated problem instances.

We make several assumption about the system for both case studies. First, we assume all volunteers enter the system during the Warm-up epoch and exit once the disaster relief effort is finished. We assume in the response phases, a volunteer responds to a task from the location of the task they responded to during previous response phase; if the volunteer does not respond to a task, then they return to their starting location. We let v_d be uniformly distributed between one and two to create variability in the importance of tasks. We let $S_d = S$ and $D_s = D$ so there are no restrictions as to which volunteers can be recommended to a task. Lastly, we assume the volunteers select tasks from the recommendations predictably but randomly. We assume a strong primacy effect occurs

due to the report that volunteers often select the first items presented to them [11]. We set the weights such that $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.125$, and so on with the last two options having equal weights. We assume volunteers select options based on the ratio the of (1) utility of the task u_{sd} and its positional weight w_i to (2) the sum of the products of the utility and positional weight of all recommended tasks. This approach is similar to that taken by Hazrati et al. [90]. Therefore, the probability of volunteer $s \in S$ selecting the task in position $i = 1, \ldots, L$ given task $d_i \in D_s$ is in position i of the recommendations r_s given to s is:

$$F_s(d_i \mid r_s) = P(s \text{ responds to } d_i) = \frac{w_i u_{sd_i}}{w_0 + \sum_{i'=1}^{L} w_{i'} u_{sd_{i'}}}$$

and zero for each task not in r_s. The value w_0 represents the "no selection" probability commonly used in the assortment optimization literature [107]. With this form, the likelihood of no selection decreases as the weighted utility of the recommendations increase. For the case studies, we present results with $w_0 = 0.1$. Limited investigation suggests the performance of the MOMM model is relatively robust to changes in this probability function. It is possible that in reality that the selection function takes another form (e.g., probability of selecting a task not recommended to them). We only introduce this function as a method to analyze the performance of the proposed methods and do not attempt to convey this as a empirically-founded function for volunteers.

Both case studies follow the logic presented in Figure 5.3 and Section 5.4.2. In the Case Study 1, there are a variable number of Response Phases as we simulate volunteer response until no tasks are left to be completed. In Case Study 2, we consider a single Response Phase. We simulate volunteer choice during each Response Phase by randomizing each volunteer's selection according to $F_s(d_i \mid r_s)$. If an active task is not completed in an epoch, it is either an active task during the next epoch or *abandons* the system. Tasks may abandon the system if a formal relief agency completes the task, the task gets aid in some other way, or the task can no longer be completed due to changes in on-the-ground conditions [119].

We assume there is a 0.25 probability of abandonment after each epoch in which the task is not served. Investigation suggests that the relative performance of the proposed methods improves as the rate of abandonment increases. At the end of each epoch, the list of active tasks and current position of each volunteer is updated and any task that was completed or abandoned the system is removed from the set of active demands.

Both case studies were run using 64 bit Python 3.7.7 on an Intel® Core™ i5-7500 CPU with 16 GB of RAM. The optimization models were solved using Gurobi 8.1.0.

5.5.1 Recommendation Models

We consider four recommendation models (k-Closest, MOMM, CM-MOMM, and T-Omniscient) to gain insight into the system and evaluate the performance of the proposed methods.

1. **k-Closest**: Recommend the k tasks closest to each volunteer. This describes the implementation of a basic recommendation systems that does not capture or consider the utility each task provides a volunteer or the societal value of the task. It also models how spontaneous volunteers may choose tasks without a recommendation system. Volunteers may review the k tasks closest to them and decide to which among those k tasks to respond, if any.

2. **MOMM**: Use the MOMM model to generate recommendations using standard objective coefficients $w_i v_d u_{sd}$. In the case studies we let $L = 3$, $J_d^\ell = 0$ for each $d \in D$, and $J_d^u = 2$ for each $d \in D$.

3. **CM-MOMM**: Use the MOMM model using coefficients generated from CM algorithm to generate recommendations. In the case studies we let $L = 3$, $J_d^\ell = 0$ for each $d \in D$, and $J_d^u = 2$ for each $d \in D$.

4. T**-Omniscient**: Assign each volunteer the first task in a route over T epochs such that the captured utility is maximized with the constraints that the highest valued tasks are completed over the T epochs, V_{max}, and each task is completed by only one volunteer. The T-Omniscient model represents a partially omniscient central controller that can assign volunteers to tasks. It provides an upper bound on the performance of any recommendation system over T epochs as the model has knowledge of the emergence and abandonment of tasks in the next $T-1$ epochs and can directly assign volunteers to tasks. Furthermore, volunteers are assumed to complete the tasks regardless of the utility it provides to them (i.e., "no selection" probability is zero).

   We present an integer programming formulation of the T-Omniscient problem. This model considers all possible routes R over the time horizon T. We allow routes that assign volunteers to return to their starting location in any epoch. We let y_{sr} be one if volunteer $s \in S$ is assigned route $r \in R$. The utility for a volunteer s to take route r is u_{sr} and is the sum of each task's utility along the route. The value of route r is v_r. Let R_d be the set of routes that contain task $d \in D$. The resulting model is a multiple vehicle routing problem with time windows. In this scenario, a time window is the epochs during which a task can be completed.

$$z^{\phi}(T) = \max \quad \sum_{r \in R} \sum_{s \in S} u_{sr}\, y_{sr} \tag{5.7}$$

$$s.t. \sum_{r \in R} \sum_{s \in S} v_r\, y_{sr} = V_{max} \tag{5.8}$$

$$\sum_{s \in S} \sum_{r \in R_d} y_{sr} \leq 1 \quad \forall\, d \in D \tag{5.9}$$

$$\sum_{r \in R} y_{sr} = 1 \quad \forall\, s \in S \tag{5.10}$$

$$y_{sr} \in \{0, 1\} \quad \forall\, s \in S,\, r \in R \tag{5.11}$$

The objective (5.7) maximizes the total utility from the assigned routes. Constraint (5.8) ensures the tasks completed provide the highest value possible. Constraint set (5.9) ensures each task is only responded to by one volunteer. Constraint set (5.10) ensures each volunteer is assigned exactly one route. Constraint set (5.11) ensures all variables are binary indicating if a route is assigned to a volunteer or not.

For large D and T, this model is difficult to solve due to the large number of possible routes that each volunteer in S can take. In reality, a central controller would not have omniscience for many, if any, time periods, so large instances would rarely be solved in practice even with a central controller.

5.5.2 Case Study 1: Hurricane Dorian

Hurricane Dorian made landfall on the Abaco Island, Bahamas as a Category 5 hurricane on September 1, 2019 [15]. One day later, it made landfall on Grand Bahamas Island still as a Category 5 hurricane [15] causing a 3.4 billion dollar impact on the Bahamas [15] with at least 70 deaths with 300 people still reported missing over 3 months after the initial landfall [95]. Soon after initial landfall, many individuals submitted a request for aid from spontaneous volunteers on the CrowdSource Rescue platform [46] for their friends, family, or even strangers they had heard were in need of aid on a different channel of communication. We use the requests collected from the CrowdSource Rescue platform as tasks in our case study.

On the Grand Bahamas Island, there were 93 unique requests made accounting for 345 adults, 124 children, 61 elderly, and 103 animals. Figure 5.4 provides the temporal pattern of request arrivals on the Grand Bahamas Island from September 2 to September 4, and Figure 5.5 provides a spatial distribution of requests. Of the requests, only 2.8% of the requests were submitted to the platform by the individual(s) needing aid. Approximately

65.7% of the tasks required a water vehicle to access the location of the task. Based on the data collected, resources to clean and curate the information would be necessary to construct a usable dataset. Approximately 22.9% of the tasks were obvious duplicates and free-fill text boxes provide valuable information about the tasks, but are difficult to deal with in an automated manner due to the lack of structure.

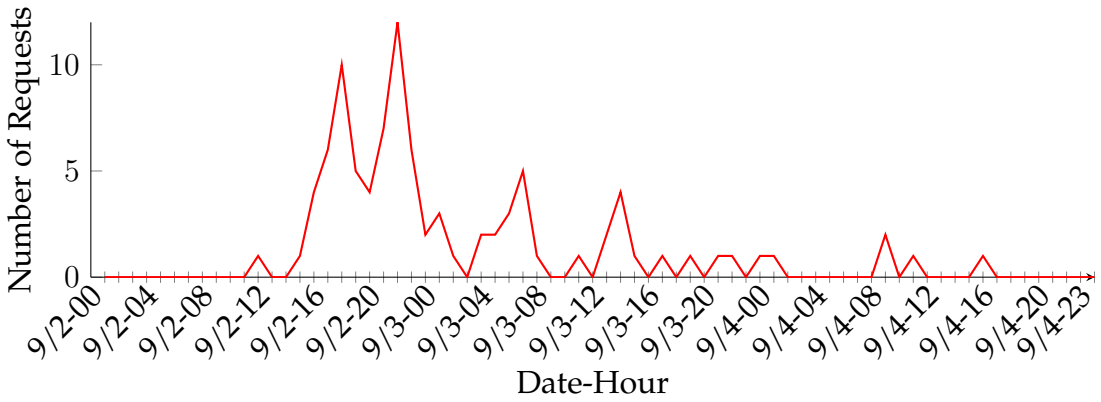


Figure 5.4: Number of new, unique request by hour on the Grand Bahamas Islands on the days during and following the landfall of Hurricane Dorian (Sept. 2, 3, 4). Horizontal axis is labeled by the date and hour of the day.

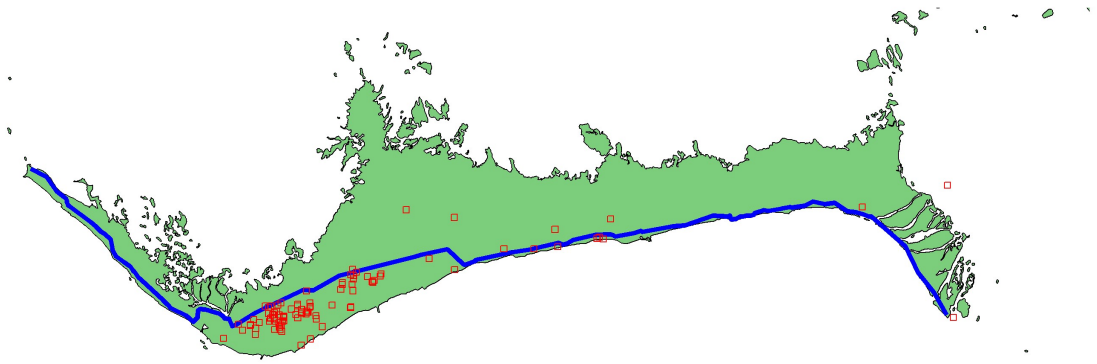


Figure 5.5: The spatial distribution of requests for aid in the days following landfall (Sept. 2, 3, 4) of Hurricane Dorian on the Grand Bahamas Islands in the Bahamas. Red squares identify estimated locations for requests of aid. The blue line represents the main highway across the island.

Experimental Design

We simulate volunteer response to tasks using Monte Carlo methods until all 93 tasks are either completed by volunteers or have abandoned the system. We assume that recommendations are updated every three hours to provide the data curators sufficient time to clean any new data, time for tasks to be completed, and time for the recommendations to be generated.

We compare six recommendation models: 1-Closest, 3-Closest, MOMM, CM-MOMM, 1-Omniscient, and 4-Omniscient. In our case study, a 17-Omniscient model would ensure the maximal number of tasks possible be served and would provide an upper bound on the performance of any recommendation system; however, the model becomes intractable due to the number of possible routes that volunteers could take. Instead, we use the 4-Omniscient model to provide a quasi-upper bound on the performance of any of the recommendation models.

We create ten disaster scenarios using the data from CrowdSource Rescue, each corresponding to a unique *abandonment profile* that describes when tasks abandon the system, the starting locations and preferences of the volunteers, and the value of the tasks. For each scenario, we run 30 replications of the response effort, since the active demands and task recommendations may change after the initial epoch depending on which task each volunteer responds to from their recommendations. Each replication consists of multiple epochs.

We consider 5 volunteers in each scenario to match the 5.4 tasks that emerge per epoch on average when the epoch length is three hours. We assume that each volunteer has a *home location* at which they begin disaster relief effort and to which they return if they are not recommended any task. We let the home locations of the volunteers be uniformly distributed along the main highway system of the Grand Bahamas Island shown in Figure

5.5 as a solid blue line. We calculate the utility each task provides each volunteer according to the following logic to capture the match between task needs and volunteers skills/preferences. Each volunteer is given a preference profile randomly. With 0.2 probability, the volunteer has an aversion to tasks dealing with pets (with 0.8 probability, the volunteer does not have an aversion). With 0.2 probability, the volunteer has an aversion to tasks dealing with children. With 0.2 probability, the volunteer has an aversion to tasks dealing with elderly. With 0.4 probability, the volunteer has an aversion to tasks dealing with a water rescue. We assume no correlation between probabilities. We calculate the utility each task provides a volunteer using this preference profile. If a task deals with an attribute (i.e., pet, child, elderly, water rescue) that the volunteer has an aversion to, the utility of that task for the volunteer is 0.001. Otherwise, each task provides a utility equal to the number of individuals (adults, children, elderly, pets) the task aids divided by the ℓ_1 (Manhattan) distance between the task and the current location of volunteer. The utilities are calculated in this way to model cases in which volunteers do not want to respond to tasks involving pets, children, elderly, or water. The 0.001 value is chosen so that a task meeting any of the conditions will provide a lower utility to the volunteer than tasks not meeting any condition for any potential distance, but if a volunteer cannot be recommended one of the higher utility tasks, a task with a utility of 0.001 is still be recommended and potentially responded to by the volunteer. Note, this utility is not intended to be a method by which to create utilities in practice. Rather, it is a method to create realistic utilities for the purpose of this case study, since we do not have information about volunteers during the disaster response efforts for Hurricane Dorian.

Experimental Results

We present the results of the simulated disaster response effort for four metrics: over-service of tasks, number of tasks completed, value of completed tasks, and the captured utility. For

each, we compare the results when using different recommendation models using paired t-tests.

Over-service of Tasks - Over-service of tasks in an indication of inefficient deployment of volunteers to disaster relief efforts. Substantial over-service can reduce reoccurring volunteer participation if the volunteers believe their efforts are not needed. Table 5.3 presents the average proportion of completed tasks responded to by one, two, three, four, or five volunteers over the ten scenarios and the average number of volunteers responding to complete each task.

We find that all models result in over 0.68 of the tasks responded to by a single volunteer. However, the 1-Closest and 3-Closest recommendations lead to 0.10 and 0.05 of the tasks responded to by three or more volunteers, respectively. The MOMM model avoids these extreme over service by using J_d^u to limit the number of volunteers that can be recommended a task. The MOMM model leads to fewer volunteers responding per task on average compared to the 1-Closest (p-value $= 0.04$) and 3-Closest ($p = 0.11$) recommendations. The CM-MOMM model improves the recommendations generated by the MOMM model by using the functions $F_s(\cdot)$ to estimate the likelihood volunteers will respond to their recommended tasks. As a result the CM-MOMM model results in significantly ($p < 0.01$) fewer number of volunteers responding to each completed task. Task may still be recommended to multiple volunteers when the value of or utility provided by the task is large and the likelihood another volunteer selects the task is low. However, due to the stochastic nature of response, multiple volunteers may choose the same task. By design, the 1-Omniscient and 4-Omniscient models can assign volunteers to tasks, so the number of volunteer responders is always one.

Table 5.3: Average proportion of tasks responded to by N volunteers for each model, and the average number of volunteers responding to a completed task over all scenarios and replications.

	Number of responders					
Model	1	2	3	4	5	Avg
1-Closest	0.68	0.21	0.08	0.02	0	1.46
3-Closest	0.77	0.17	0.04	0.01	0	1.29
MOMM	0.76	0.24	0	0	0	1.24
CM-MOMM	0.83	0.17	0	0	0	1.17
1-Omniscient	1.0	0	0	0	0	1
4-Omniscient	1.0	0	0	0	0	1

Number of Completed Tasks - Two other key metrics of the OTR problem are the number of tasks completed by the spontaneous volunteers and how quickly these tasks are completed after being submitted to the platform. Figure 5.6 illustrates the average cumulative number of tasks completed within the number of epochs indicated by the horizontal axis after being submitted to the platform. The number of epochs is calculated as the number of response phases between being added to the platform and being completed by a volunteer, which is at least one, since the task will not be recommended until the construction phase following being added to the platform. The steeper the curve and the higher the vertical value, the better as this indicates that more tasks are being completed within a low number of epochs after initial request and that more tasks are being completed before abandonment. The final entry $(20^{th}$ epoch$)$ represents the number of tasks completed during the simulated disaster response efforts.

We find that the MOMM model completes a significantly larger number of tasks (44.2) than the 1-Closest (31.2) and 3-Closest (37.2) recommendations at levels $p < 0.01$ and $p < 0.01$, respectively. This is because the MOMM model promotes high-utility tasks, so volunteers are more likely to participate during each epoch. We also find the CM-MOMM

recommendations lead to more completed tasks (46.0) when compared to the MOMM model ($p < 0.01$). The CM-MOMM model recommendations also lead to more completed tasks more quickly. The CM-MOMM model does this by reducing the over-service of tasks so that volunteers are the only one to respond to and complete requests more often.

Both the 1-Omniscient (55.7) and 4-Omniscient (62.9) result in a larger number of completed tasks as they are able to directly assign volunteers to tasks and as it is assumed every volunteer will participate in every epoch, regardless of the utility a task provides them. The 4-Omniscient uses the knowledge of future task abandonment and ability to delay the completion of high-value tasks until subsequent epochs to increase the number of tasks completed.

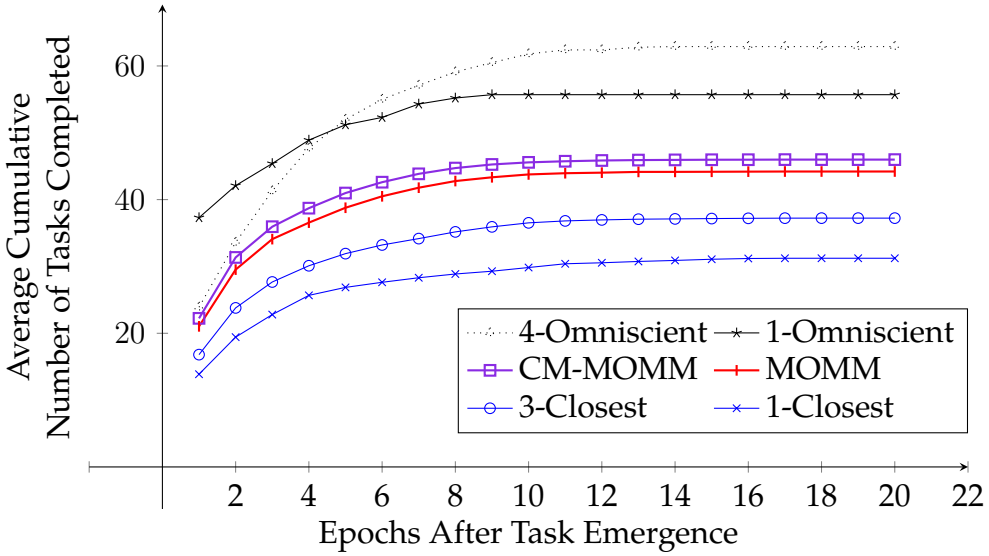


Figure 5.6: Average cumulative number of tasks completed within the number of epochs on horizontal axis from when first added to the system across the 10 scenarios.

Value of Completed Tasks - It is also critical that the high value tasks are completed before abandoning the system. Table 5.4 presents the average total value of completed tasks for the ten disaster relief scenarios. We find that the MOMM recommendations result in a higher total value of completed tasks when compared to the 1-Closest and 3-Closest

models ($p < 0.01$ and $p < 0.01$ respectively). This is a result of including the value of the recommended task as a component of the MOMM objective function coefficient. Neither the 1-Closest model nor the 3-Closest model account for the value of tasks. We also find that the CM-MOMM model increases the average total value completed tasks compared to the MOMM model ($p < 0.01$). This is a result of the lower over-service offered by the CM-MOMM recommendations.

The 1-Omniscient and 4-Omniscient models lead to a higher average total value of completed tasks as they can directly assign volunteers to tasks to reduce over-service and ensure the volunteers complete the highest value tasks. When there are tasks that are of high value but of low utility to each volunteer it is unlikely the tasks will be completed when using the MOMM and CM-MOMM models. While not considered in this case study, in practice J_d^ℓ can be increased for high value tasks that are not being completed causing the task to be recommended to more volunteers. Another approach is to progressively increase the value of uncompleted tasks each epoch so the task is placed in higher weight (w_i) positions.

Table 5.4: Average total value of completed tasks over 30 replications.

	Average Total Value of Completed Tasks					
Scenario	1-Closest	3-Closest	MOMM	CM-MOMM	1-Omniscient	4-Omniscient
0	48.19	54.7	68.98	72.2	93.94	101.07
1	56.2	65.28	68.81	72.43	88.94	102.72
2	50.17	55.31	67.99	72.14	92.24	98.35
3	43.53	51.63	65.48	68.41	88.87	100.43
4	59.07	69.54	72.05	75.12	90.82	102.9
5	32.45	42.9	62.74	63.17	89.94	102.4
6	48.8	60.09	68.04	71.49	91.02	101.8
7	34.97	40.6	49.64	49.7	87.88	100.11
8	53.79	59.34	68.19	70.71	88.06	101.41
9	36.5	55.13	75.55	78.18	92.27	102.35
Average	46.36	55.45	66.75	69.36	90.4	101.35

Captured Utility - Lastly, it is critical to ensure volunteers gain utility from completing tasks to promote reoccurring participation by volunteers. The amount of utility captured for each completed task is equal to the highest utility among all volunteers responding to that task. Table 5.5 presents the average amount of utility captured in each scenario over 30 replications for each scenario. We find the MOMM model significantly outperforms the 1-Closest ($p < 0.01$) and the 3-Closest ($p < 0.01$) recommendations. This is because the MOMM model directly incorporates the platform's estimate for the utility when generating the recommendations. With the additional information of the functions $F_s(\cdot)$ for the volunteers, the CM-MOMM model outperforms the MOMM model with a p-value of 0.26. Compared to the number of completed tasks and total value of completed tasks, the captured utility is relatively similar between the MOMM and CM-MOMM recommendations. The can be explained by the multi-epoch nature of the disaster response effort. As discussed earlier, the CM-MOMM model is able to complete more tasks in an epoch on average. This leaves fewer tasks left to complete in subsequent time epochs. All things else being equal, if there are more tasks in the system, then there are more tasks to recommend, and some may provide a relatively high utility to some volunteers compared to the utility captured by volunteers in earlier epochs. Using the case study in Section 5.5.3, we show the CM-MOMM recommendations have a higher expected utility in a single epoch.

We find that the MOMM and CM-MOMM recommendations result in a higher captured utility than the 1-Omniscient central controller ($p < 0.01$ and $p < 0.01$ respectively). This is due to the requirement that the 1-Omniscient capture the largest amount of value in each epoch as possible. The central controller is forced to assign volunteers to tasks that may be relatively undesirable for the volunteer, but societally beneficial. The MOMM and CM-MOMM models results in a delayed completion of high-valued tasks, but captures a higher utility in doing so. Having the volunteers complete tasks that provide a high utility is critical to reoccuring participation [162]. We find that a knowledge of system states in

future epochs allows the 4-Omniscient model to lead to a larger amount of utility captured compared to the MOMM ($p = 0.11$) and CM-MOMM ($p = 0.31$) by strategically assigning volunteers to tasks to ensure the best volunteer completes each task, to avoid over-service, and to complete more tasks.

Table 5.5: Average utility captured over 30 replications.

	Average Utility Captured					
Scenario	1-Closest	3-Closest	MOMM	CM-MOMM	1-Omniscient	4-Omniscient
0	306.41	395.73	555.12	567.54	161.6	545.12
1	265.52	447.91	572.9	582.57	369.15	779.93
2	556.6	497.32	468.97	503.05	177.69	640.35
3	185.35	228.75	423.67	418.52	163.79	414.57
4	420.25	485.72	607.86	614.44	307.45	573.29
5	87.43	77.24	245.09	222.6	207.3	369.19
6	454.98	506.68	576.04	614.04	307.31	966.47
7	234.38	283.78	326.34	316.94	182.32	454.29
8	405.8	569.18	587.56	611.07	243.38	775.6
9	80.9	178	395.45	438.23	154.15	434.25
Average	299.76	367.03	475.9	488.9	227.41	595.31

5.5.3 Case Study 2: Large Instances

We investigate the proposed models for various sized problem instances to understand how the performance of the models and computational times scale. We compare five recommendation models: 1-Closest, 3-Closest, MOMM, CM-MOMM, and 1-Omniscient. Since we only consider a single epoch, we do not solve a 4-Omniscient model.

Synthetic Data Generation

We create synthetic data sets to mimic how tasks may be spatially distributed during a disaster relief effort. It is likely that tasks with similar features, such as the need for a water rescue or serving elderly individuals, would be spatially located near each other.

To capture this, we assume tasks are spatially distributed according to a two-dimensional Gaussian mixture model with ten equally likely distributions. The mean of each Gaussian distribution is uniformly randomized between zero and one. We set the coefficient matrix of each Gaussian distribution to a diagonal matrix with a random value between 0 and 0.04 on the diagonal to produce unique distributions with varying spread. We refer to the tasks generated from the same Gaussian distribution as a cluster. The location of volunteers are uniformly randomized on the two dimensional plane between zero and one. Illustrations of tasks (●) and volunteers (■) created using this method are provided in Figure 5.7.

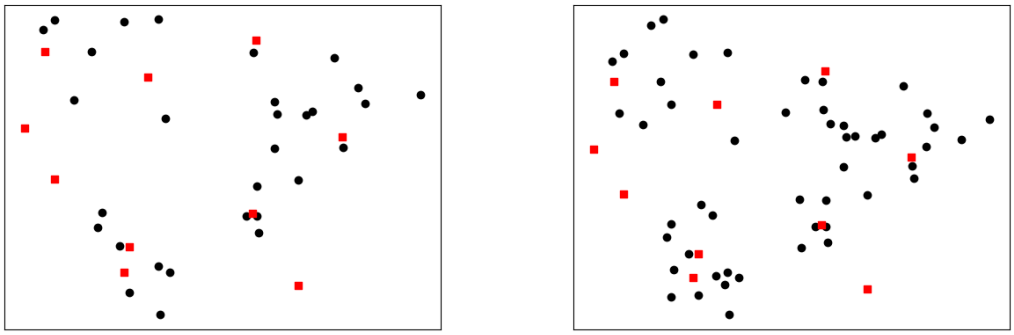


Figure 5.7: Tasks (●) and volunteers (■) generated by the above procedure for $|D| = 30,\ |S| = 10$ (left) and $|D| = 50,\ |S| = 10$ (right).

We generate the utilities such that the utilities for each volunteer and task are varied but, given a volunteer, there is a correlation between the utility of tasks from the same cluster and, given a task, there is a correlation between the utility each volunteers has of completing it. We separate the utility calculation into different components. First, each task has a base level utility component. For example, this could represent the number of individuals who would benefit from the task being completed. We denote this utility component as u_d for each task $d \in D$. We set u_d uniformly between zero and one. Second, a volunteer has a utility component relating to tasks from a given cluster, since we assume the tasks in a cluster have similar features. We denote this utility component as u_{sc_d} for each

volunteer $s \in S$ and cluster c_d of task $d \in D$. We set u_{sc_d} uniformly between zero and one. Lastly, a volunteer has a unique utility component for each task relating to unique features of the task. We denote this utility component as u'_{sd} for each volunteer $s \in S$ and task $d \in D$. We set u'_{sd} uniformly between zero and one. We also let the utility depend on the distance between the volunteer $s \in S$ and the task $s \in D$. We consider utility to be inversely proportional to the ℓ_1 distance between them, $\ell_1(s, d)$. The utility u_{sd} of volunteer $s \in S$ completing task $d \in D$ used in the case study is subsequently calculated as $u_{sd} = \frac{u_{sc_d} u'_{sd} u_d}{\ell_1(s,d)}$ for each $s \in S,\ d \in D$. Note, this utility is not intended to be a method by which to create utilities in practice.

Experimental Results

Results are analyzed for varying sized instances with respect to three metrics: solution time, expected total value of completed tasks, and expected utility captured.

Solution Time - Table 5.6 presents the number of seconds required for each method to generate recommendations for instances of various sizes. The MOMM and CM-MOMM models are solved using the standard linear programming relaxation of the integer program formulation. The MOMM and CM-MOMM models solve within 77.1 and 173.1 seconds, respectively, across all instances. Algorithms using the minimum cost network flow formulation took longer to solve, so results are not presented. Low solution times for even relatively large problem instances suggests that such an approach can be used to generate recommendations in real time for an online platform.

Expected Total Value of Completed Tasks - Table 5.7 presents the expected value of completed tasks from the recommendations generated for each problem instance. For all problem instances, we find that the MOMM recommendations lead to an expected value of completed tasks larger than the 1-Closest and 3-Closest models. For all problem instances, the CM-MOMM recommendations have at least as large an expected value of completed

Table 5.6: The seconds required to generate recommendations using each recommendations system. The MOMM and CM-MOMM models were solved using the standard linear programming relaxation of the integer program formulation.

		Solution Time (seconds)				
$\|S\|$	$\|D\|$	1-Closest	3-Closest	MOMM	CM-MOMM	1-Omniscient
10	10	0.00	0.00	0.01	0.01	0.06
	30	0.00	0.00	0.02	0.03	0.02
	50	0.00	0.00	0.02	0.04	0.03
	100	0.00	0.00	0.05	0.08	0.06
	150	0.00	0.00	0.08	0.12	0.07
50	50	0.00	0.00	0.1	0.24	0.13
	150	0.02	0.01	0.6	1.15	0.37
	250	0.03	0.03	0.78	1.64	0.67
	500	0.1	0.1	2.32	4.12	1.57
	750	0.21	0.21	3.23	5.41	2.75
100	100	0.01	0.01	0.39	1.02	0.43
	300	0.07	0.07	2.18	5.66	1.83
	500	0.18	0.18	4.71	9.39	3.23
	1000	0.71	0.71	13.09	21.75	8.89
	1500	1.44	1.44	16.37	27.42	16.96
300	300	0.21	0.21	4.81	22.04	6.16
	500	0.54	0.53	16.96	43.88	14.11
	1000	2.04	2.07	48.05	119.35	35.79
	1500	4.36	4.35	77.1	173.1	62.89

tasks as the MOMM model. Due to the ability to directly assign volunteers to tasks, the 1-Omniscient model has a higher expected value from completed tasks.

Expected Utility Captured - Table 5.8 presents the expected utility from the recommendations generated for each problem instance. For all problem instances, the MOMM recommendations have a larger expected utility captured than the 1-Closest and 3-Closest models. The CM-MOMM recommendations have a larger expected utility captured than the MOMM model for all but one instance. By design, the CM-MOMM model re-weights different matchings to promote a higher expected utility captured. As found before, the 1-Omniscient model results in lower expected utility as volunteers are assigned to tasks that provide the

highest societal value, but may be relatively undesirable for the volunteers assigned to the task. The MOMM and CM-MOMM models are able to recommend tasks that may provide lower societal value, but results in more utility captured by the volunteers. This is critical to promote reoccurring participation by the volunteers [162].

Table 5.7: The expected value of completed tasks using each recommendations system during the single epoch considered.

		Expected Value of Completed Tasks				
$\|S\|$	$\|D\|$	1-Closest	3-Closest	MOMM	CM-MOMM	1-Omniscient
10	10	7.11	7.01	9.23	9.23	16.42
	30	10.06	9.66	11.08	11.51	18.05
	50	10.25	10.23	12.28	13.17	19.02
	100	12.69	12.62	14.06	14.6	19.5
	150	12.04	11.86	13.21	13.63	19.61
50	50	34.37	37.04	46.82	48.18	77.56
	150	43.94	47.86	60.53	63.94	90.24
	250	49.69	52.71	65.72	70.25	83.58
	500	56.98	59.98	67.09	72.34	81.97
	750	56.94	60.22	68.77	73.09	96.23
100	100	75.28	78.56	104.02	106.87	152.8
	300	100.48	103.94	122.45	134.14	178.04
	500	107.74	113.54	127.59	143.57	175
	1000	105.17	113.03	143.42	153.13	190.61
	1500	120.49	130.65	142.14	152.38	187.7
300	300	173.48	186.37	292.2	297.86	448.98
	500	222.69	229.12	338.77	357.09	487.61
	1000	273.19	286.78	378.5	410.34	537.36
	1500	304.56	313.67	388.01	427.22	538.77

Table 5.8: The expected amount of utility captured using each recommendations system during the single epoch considered.

		Expected Utility Captured				
$\|S\|$	$\|D\|$	1-Closest	3-Closest	MOMM	CM-MOMM	1-Omniscient
10	10	5.51	5.17	6.69	6.69	7.85
	30	8.43	8.13	10.94	11.06	9.01
	50	14.19	13.12	15.46	15.54	8.82
	100	33.33	32.87	35.38	35.86	11.2
	150	37.6	36.84	40.77	41.35	11.36
50	50	69.4	74.17	90.61	92.86	109.11
	150	132.69	152.6	171.34	176.96	120.79
	250	123.58	164.11	202.99	208.27	99.96
	500	279.92	316.12	365.6	370.21	88.8
	750	303.6	348.17	409.34	415.41	86.99
100	100	349.99	370.46	395.63	398.41	467.29
	300	572.65	623.31	672.87	698.5	383.82
	500	628.49	741.65	811.82	839.86	344.82
	1000	2183.03	2191.18	2295.48	2320.73	577.2
	1500	1004.4	1175.82	1281.35	1306.14	336.41
300	300	964.7	1043.22	1224.58	1230.99	1439.79
	500	1182.52	1301.92	1543.64	1572.79	1481.22
	1000	1704.9	2029.92	2371.4	2436.22	1383.97
	1500	2421.91	2763.96	3169.38	3246.49	1680.72

5.6 Conclusions

Spontaneous, self-assigning volunteers are critical resources during disaster response efforts. However, they often act in an uncoordinated fashion which leads to inefficient response efforts. Systems can be put in place to better manage the spontaneous, self-assigning volunteer resources during disaster relief.

This chapter serves as a proof of concept for a framework to provide task recommendations to volunteers. We present the MOMM model as a heuristic to generate recommendations within the Online Task Recommendation problem framework when knowledge

of key system components is limited. We then present the coefficient modification algorithm to improve the recommendations generated by the MOMM model in the instances where we can estimate the probability a volunteer would select specific tasks from their recommendations. Using case studies we demonstrate the benefit of both models.

The Online Task Recommendation problem represents a research direction rich in opportunities. First, techniques are needed to predict the emergence of tasks to the platform. Second, techniques are needed to estimate or elicit volunteers' preference for various tasks before recommendations are generated. Third, it is necessary to better understand the interaction of spontaneous volunteers with the platform and identify functions to describe how they select tasks from their recommendations. Developing these techniques will allow the complex stochastic, multi-epoch nature of the problem to be incorporated into the recommendation construction process and lead to improved recommendations. These technologies would facilitate improved coordination of actors within the system, improve social welfare, and improve the safety of individuals following a disaster incident.

Chapter 6

Final Remarks

The research included in this dissertation investigates unstudied problems related to election voting system management and proposes new operations research approaches for these problems. Through the research, we demonstrate the proposed approaches can help election officials plan election systems that are more resilient, cost-effective, and accessible. In this concluding chapter, I summarize the key results from each primary chapter and discuss the value of our research. I then review the limitations of our research and propose directions for future research.

In Chapter 2, we study the impact of pandemic-related voting system disruptions on key voter metrics and assess proposed mitigating practices using a discrete event simulation of the in-person voting system. We find that poll worker shortages, social distancing, personalized protective equipment use, and sanitation measures can lead to extremely long voter wait times. We also find that long wait times can be avoided by staffing additional check-in locations, expanding early voting, and increasing the physical footprint of polling locations.

In Chapter 3, we propose and study an integer programming model to identify when and how to consolidate polling locations. We prove that the polling location consolidation

problem is theoretically challenging, which supports the use of optimization for this problem. Using a case study, we outline implications for practice. Most notably, we find that consolidating a moderate number of polling locations may be a reasonable approach to mitigating disruptions to the voting system. We also demonstrate the value of strategically located, large back-up buildings that can be used as polling locations when consolidation is necessary.

In Chapter 4, we propose and study an integer programming model to select ballot drop box locations that balance the trade-off between cost, voter access, and risk. The DBLP is shown to be NP-Hard, and we introduce a heuristic to generate a large number of feasible solutions for policy makers to select from a posteriori. Using a real-world case study of Milwaukee, WI, we demonstrate that the proposed optimization model identifies drop box locations that perform well across multiple criteria and outperforms naive rules-of-thumb.

In Chapter 5, we propose a framework and integer program, the MOMM, to construct task recommendations for spontaneous actors with free choice. While originally designed for use within disaster response efforts, the framework and model can be used to indirectly control actors within the election system to reduce the "cost" of voting experienced by potential voters. We prove the MOMM is polynomially solvable, and we present a method to calculate improved objective coefficients by iteratively solving the proposed integer program and computing the likelihood that actors would select each recommended action. Using a case study, we demonstrate that recommendations generated by the proposed models improve the actions taken by self-assigning actors.

The contents and contributions of Chapter 5 can also be adapted for use when more substantial changes are made to the in-person voting infrastructure. Some jurisdictions may elect to reduce the number of polling locations offered and allow voters to cast a ballot at any of the locations, referred to as *voting centers* [185]. However, there may be some voting centers that are preferred by a large number of voters. When voters can freely select

the voting centers at which they vote, congestion and long wait times are likely to occur. We posit that we can recommend each voter a small number of voting centers at which to vote in a way that improves the system-wide performance of the voting system while still allowing voters to select the center at which they vote. The recommendations can be strategically generated to balance the distance between the voter, their available modes of transportation, and the wait time they ultimately experience once arriving at their polling location.

The research contained in each of these chapters demonstrate the importance and value of operations research to elections-related problems. In addition to providing value to the election system, the research contained within this dissertation contributes new knowledge to the field of industrial engineering. In each chapter, we introduce new mathematical models of complex inter-related voting systems. We then study the theoretical and practical implications of those models. We introduce new algorithms to support the use of the models and to elicit information about the voting systems. However, the techniques introduced in each chapter represent only the initial attempt at investigating these problems. A major contribution of this dissertation is the mathematical frameworks of the election voting system that we introduce. The mathematical frameworks will encourage and support future research by industrial engineers into the election voting system.

As with all research studies, there are limitations to our contributions and the applicability of our results to real-world voting systems. First, to provide a tractable mathematical models of the voting systems we study, we make simplifying assumptions. The specific assumptions of each model is discussed in the respective chapters. Despite these simplifications, we demonstrate the value of the models through our research. There are also limitations to the generalizability of the statements made about practice or policy. We consider only a small number of case-studies in our research and we only consider "typical" voting systems. Despite this, the models were designed with multiple states and

jurisdictions in mind so that the models and the practical implications should be generally applicable to other voting systems. Another limitation of our work is that the proposed models and their corresponding recommendations were only studied computationally and were not implemented in practice. This limits the extent to which we can assess the true value of the proposed models.

The contributions of this dissertation address only a small number opportunities for the operations research community to support the design of resilient election voting systems. I outline potential directions for future work. First, this dissertation does not address all potential actions that election administrators can take in response to a disruption. For example, there may more dramatic changes to the voting infrastructure, such as converting from traditional polling locations to voting centers among which any voter can chose to vote. Deciding the location of voting centers is complex, since decisions made by individual voters are not always in the best interest of the entire system. Placing voting centers near a large number of voters can lead many voters to vote there, but this can increase the congestion and wait time experienced once arriving at the polling location. In this way, there is a trade-off between voting-related "costs" that must be carefully analyzed when selecting the location of voting centers. There are also operational decisions that can improve the resiliency of voting systems that have not been addressed within this dissertation or the literature. For example, jurisdictions may withhold critical voting resources (e.g., voting machines) from initial allocation and instead deploy the withheld resources during the election once information about voter turnout at each polling location becomes available. Research is needed to understand how many resources to withhold from initial allocation, when and where to deploy resources during the election, and how to predict the number of voter arrivals over the course of the Election Day.

Second, the models introduced within this dissertation only consider ways to redesign the voting system for a single upcoming election. However, some decisions have longer

lasting impacts. For example, many ballot drop boxes are permanent and remain in the same location for multiple elections. Future work is needed to build analytical tools that can use projections about future election periods and changing voter dynamics to inform decisions.

Third, research is needed to develop a mathematical model of the voting system and voter participation that considers all possible voting system design decisions, examines their interactions, and identifies a cost-effective portfolio of actions for election officials to take. An important challenge of this problem is incorporating models to predict the choices made by voters, reliability uncertainty, and recourse decisions (i.e, real-time voting machines deployment) since those choices ultimately determine the cost-effectiveness of the voting system.

Finally, many of the proposed optimization models do not always provide intuitive policies for election administrators to use in practice. Moreover, the proposed models may be too mathematically-intensive for election administrators to use directly. This may hinder the acceptance and use of the models in practice. Additional research is needed to develop simple, easy-to-implement policies that can be used by election officials. Accordingly, industrial engineers have a large future role in the design of effective, equitable, and resilient voting systems.

Bibliography

[1] 42 U.S. Code § 5195c - Critical infrastructures protection. URL https://www.law.cornell.edu/uscode/text/42/5195c.

[2] 2019 Act No. 54 (R74, H3035) South Carolina Code of Laws, 2019.

[3] Louisiana Cajun Navy, 2019. URL https://www.facebook.com/LaCajunNavy/.

[4] Hussain Abualkhair, Emmett J. Lodree, and Lauren B. Davis. Managing volunteer convergence at disaster relief centers. *International Journal of Production Economics*, page 107399, July 2019. ISSN 0925-5273. doi: 10.1016/j.ijpe.2019.05.018. URL http://www.sciencedirect.com/science/article/pii/S0925527319301999.

[5] Amir Ahmadi-Javid, Pardis Seyedi, and Siddhartha S. Syam. A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263, March 2017. ISSN 0305-0548. doi: 10.1016/j.cor.2016.05.018.

[6] Theodore Allen and Mikhail Bernshteyn. Mitigating Voter Waiting Times. *CHANCE*, 19:25–34, August 2006. doi: 10.1080/09332480.2006.10722812.

[7] Foued Aloulou. *The Application of Discrete Choice Models in Transport*. IntechOpen, November 2018. ISBN 978-1-78984-396-5. doi: 10.5772/intechopen.74955. URL https://www.intechopen.com/books/statistics-growing-data-sets-and-growing-demand-for-statistics/the-application-of-discrete-choice-models-in-transport. Publication Title: Statistics - Growing Data Sets and Growing Demand for Statistics.

[8] Nezih Altay and Walter G. Green. OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475–493, November 2006. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.05.016. URL http://www.sciencedirect.com/science/article/pii/S0377221705005047.

[9] Sardar Ansari, Laura Albert McLay, and Maria E. Mayorga. A Maximum Expected Covering Problem for District Design. *Transportation Science*, 51(1):376–390, February 2017. ISSN 0041-1655. doi: 10.1287/trsc.2015.0610. URL http://pubsonline.informs.org/doi/abs/10.1287/trsc.2015.0610. Publisher: INFORMS.

[10] Masaki Aoyagi and Seung Han Yoo. Matching Platforms. SSRN Scholarly Paper ID 3514317, Social Science Research Network, Rochester, NY, December 2019. URL https://papers.ssrn.com/abstract=3514317.

[11] Daniel Auferbauer, Roman Ganhör, and Hilda Tellioglu. Moving Towards Crowd Tasking for Disaster Mitigation. In *12th International Conference on Information Systems for Crisis Response and Management*, May 2015.

[12] Daniel Auferbauer, Roman Ganhör, Hilda Tellioglu, and Jasmin Pielorz. Crowdtasking: Field Study on a Crowdsourcing Solution for Practitioners in Crisis Management. In *13th International Conference on Information Systems for Crisis Response and Management*, May 2016.

[13] Roberto Baldacci, Marco Boschetti, Vittorio Maniezzo, and Marco Zamboni. Scatter Search Methods for the Covering Tour Problem. volume 30, pages 59–91. January 2005. ISBN 978-1-4020-8134-7. doi: 10.1007/0-387-23667-8_3.

[14] Abigail Becker. Madison crushes early in-person voting record. *The Cap Times*, August 2018. URL https://madison.com/news/local/govt-and-politics/madison-crushes-early-in-person-voting-record/article_f861d0ca-6415-59ba-8498-90c99bf6acde.html. Accessed on Aug. 6, 2020.

[15] Omar Bello, Luciana Fonted de Meira, Candice Gonzales, Leda Peralta, Nyasha Skerette, Blaine Marcano, Machel Patin, Gabriel Vivas, Carlos Espiga, Eduardo Allen, Esteban Ruiz, Francisco Ibarra, Francklin Espiga, Mauricio Gonzalez, Salvador Marconi, Daniela Carrera, Michael Nelson, Yuri Chakalall, Alejandra Perroni, Christopher Persaud, Chitralekha Deopersad, and Malaika Mason. Assessment of the Effects and Impacts of Hurricane Dorian in the Bahamas. Technical report, Inter-American Development Bank, August 2020. URL https://publications.iadb.org/en/node/28642. Edition: 2020.

[16] Nicholas D. Bernardo, Gretchen A. Macht, and Jennifer Lather. Analysis of Layout Impacts on Resource Allocation for Voting: A Los Angeles Vote Center. In *2020 Winter Simulation Conference (WSC)*, pages 3128–3139, Orlando, FL, USA, December 2020. IEEE. ISBN 978-1-72819-499-8. doi: 10.1109/WSC48552.2020.9383877. URL https://ieeexplore.ieee.org/document/9383877/.

[17] Chawis Boonmee, Mikiharu Arimura, and Takumi Asada. Facility location optimization model for emergency humanitarian logistics. *International Journal of Disaster Risk Reduction*, 24:485–498, September 2017. ISSN 2212-4209. doi: 10.1016/j.ijdrr.2017.01.017.

[18] Henry E. Brady and John E. McNulty. Turning Out to Vote: The Costs of Finding and Getting to the Polling Place. *American Political Science Review*, 105(1):115–134, 2011.

[19] Brennan Center for Justice. Voting Laws Roundup: May 2021, May 2021. URL https://www.brennancenter.org/our-work/research-reports/voting-laws-roundup-may-2021. Accessed on Aug. 22, 2021.

[20] D. A. Buell, E. Hare, F. Heindel, C. Moore, and B. Zia. Auditing a DRE-based election in South Carolina. In *Proceedings of the 2011 conference on Electronic voting technology/workshop on trustworthy elections*, EVT/WOTE'11, page 11, USA, August 2011. USENIX Association.

[21] Duncan A. Buell. An Analysis of Long Lines in Richland County, South Carolina. In *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, 2013. URL https://www.usenix.org/conference/evtwote13/workshop-program/presentation/buell.

[22] Barry Burden. Expert testimony. United States District Court, Western District of Wisconsin, Case: 3:20-cv-00249-wmc, Document #: 418, July 8, 2020.

[23] Barry Burden. The Experiences of Municipal Clerks and the Electorate in the November 2020 General Election in Wisconsin. Technical report, University of Wisconsin - Madison, 2021. URL https://elections.wisc.edu/2020-wisconsin-election-report/.

[24] Jason Calvi. 'Silliest thing that ever happened:' After long waits to vote, we'll wait 6 days for results. *FOX 6 Now Milwaukee*, April 2020. URL https://www.fox6now.com/news/silliest-thing-that-ever-happened-after-long-waits-to-vote-well-wait-6-days-for-results. Accessed on Aug. 6, 2020.

[25] Enrico Cantoni. A precinct too far: Turnout and voting costs. *American Economic Journal: Applied Economics*, 12(1):61–85, January 2020. doi: 10.1257/app.20180306. URL https://www.aeaweb.org/articles?id=10.1257/app.20180306.

[26] Enrico Cantoni. A Precinct Too Far: Turnout and Voting Costs. *American Economic Journal: Applied Economics*, 12(1):61–85, January 2020. ISSN 1945-7782. doi: 10.1257/app.20180306. URL http://www.aeaweb.org/articles?id=10.1257/app.20180306.

[27] Aakil M. Caunhye, Xiaofeng Nie, and Shaligram Pokharel. Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences*, 46 (1):4–13, March 2012. ISSN 0038-0121. doi: 10.1016/j.seps.2011.04.004. URL http://www.sciencedirect.com/science/article/pii/S0038012111000176.

[28] CBS 58 Newsroom. City of Milwaukee names five in-person voting locations, 2020. URL https://www.cbs58.com/news/city-of-milwaukee-names-five-in-person-voting-locations. Accessed on Aug. 4, 2020.

[29] Centers for Disease Control and Prevention. Considerations for Election Polling Locations and Voters, April 2020. URL https://www.cdc.gov/coronavirus/2019-ncov/community/election-polling-locations.html. Accessed on Sept. 8, 2020.

[30] Centers for Disease Control and Prevention. People with Certain Medical Conditions. Accessed on Aug. 5, 2020, 7 2020. URL https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html.

[31] W.-C. Chan and Y.-B. Lin. Waiting time distribution for the m/m/m queue. *IEE Proceedings - Communications*, 150(3):159, 2003. ISSN 13502425. doi: 10.1049/ip-com:20030274.

[32] Bassam H. Chaptini and S. Lerman. *Use of discrete choice models with recommender systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[33] M. Keith Chen, Kareem Haggag, Devin G Pope, and Ryne Rohla. Racial Disparities in Voting Wait Times: Evidence from Smartphone Data. Working Paper 26487, National Bureau of Economic Research, November 2019. URL http://www.nber.org/papers/w26487. Series: Working Paper Series.

[34] Peng Cheng, Xiang Lian, Zhao Chen, Rui Fu, Lei Chen, Jinsong Han, and Jizhong Zhao. Reliable Diversity-Based Spatial Crowdsourcing by Moving Workers. *Proceedings of the VLDB Endowment*, 8(10):1022–1033, June 2015. ISSN 21508097. doi: 10.14778/2794367.2794372. URL http://arxiv.org/abs/1412.0223. arXiv: 1412.0223.

[35] Anand Inasu Chittilappilly, Lei Chen, and Sihem Amer-Yahia. A Survey of General-Purpose Crowdsourcing Techniques. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2246–2266, September 2016. ISSN 2326-3865. doi: 10.1109/TKDE.2016.2555805.

[36] Richard Church and Charles ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32(1):101–118, December 1974. ISSN 1435-5957. doi: 10.1007/BF01942293. URL https://doi.org/10.1007/BF01942293.

[37] City of Milwaukee Common Council. Final voting ward demographics. Accessed on Aug. 4, 2020, 2012. URL https://city.milwaukee.gov/ImageLibrary/Groups/ccCouncil/2012PDF/FinalVotingWardDemographics-Ju.xls.

[38] City of Milwaukee Open Data Portal. 2020 General Election - City of Mlwaukee Canvassed Election Results. Technical report. URL https://data.milwaukee.gov/dataset/2020-general-election/resource/00b02d77-d24a-4d80-a5f0-adb2d58946db.

[39] Justin Colannino, Mirela Damian, Ferran Hurtado, Stefan Langerman, Henk Meijer, Suneeta Ramaswami, Diane Souvaine, and Godfried Toussaint. Efficient Many-To-Many Point Matching in One Dimension. *Graphs and Combinatorics*, 23(1):169–178, June 2007. ISSN 1435-5914. doi: 10.1007/s00373-007-0714-3. URL https://doi.org/10.1007/s00373-007-0714-3.

[40] Loren Collingwood, William McGuire, Benjamin Gonzalez O'Brien, Katherine Baird, and Sarah Hampson. Do Drop Boxes Improve Voter Turnout? Evidence from King County, Washington. *Election Law Journal: Rules, Politics, and Policy*, 17(1):58–72, February 2018. ISSN 1533-1296. doi: 10.1089/elj.2017.0450. URL http://www.liebertpub.com/doi/full/10.1089/elj.2017.0450. Publisher: Mary Ann Liebert, Inc., publishers.

[41] Election Assistance Commission. Election Administration And Voting Survey 2018 Comprehensive Report. A Report to the 116th U.S. Congress, 1335 East West Highway, Suite 4300 Silver Spring, MD 20910, 2018.

[42] William J. Cook, William H. Cunningham, William R. Pulletblank, and Alexander Schrijver. *Combinatorial Optimization*. Wiley-Interscience ser9es in discrete mathematics and optimization. Wiley-Interscience Publication, 605 Third Avenue, New York, NY 10158-0012, 1998.

[43] Damon P. Coppola. *Introduction to International Disaster Management*. Elsevier Science & Technology, Oxford, UNITED STATES, 2015. ISBN 978-0-12-801703-6. URL http://ebookcentral.proquest.com/lib/wisc/detail.action?docID=1929997.

[44] Nick Corasaniti, Michael D. Shear, and Trip Gabriel. Postal Crisis Has States Looking for Alternatives to Mail-In Ballots. *The New York Times*, August 2020. ISSN 0362-4331. URL https://www.nytimes.com/2020/08/17/us/politics/postal-service-voting.html.

[45] Corporation for National & Community Service. Managing Spontaneous Volunteers. URL https://www.nationalservice.gov/sites/default/files/olc/moodle/ds_managing_volunteers_in_time/viewbfff.html?id=3198&chapterid=1924. Accessed on Nov. 25, 2019.

[46] CrowdSource Rescue. URL https://crowdsourcerescue.com/. Accessed on Dec. 3, 2019.

[47] Brendan Cullerton. 'It's almost embarrassing:' Milwaukee city leaders pin blame on all sides for election day chaos. *CBS*, April 2020. URL https://www.cbs58.com/news/its-almost-embarrassing-milwaukee-city-leaders-pin-blame-on-all-sides-for-election-day-chaos. Accessed on Aug. 6, 2020.

[48] John R. Current and David A. Schilling. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, 73(1):114–126, February 1994. ISSN 0377-2217. doi: 10.1016/0377-2217(94)90149-X. URL http://www.sciencedirect.com/science/article/pii/037722179490149X. Publisher: North-Holland.

[49] Dharma Dailey and Kate Starbird. Social Media Seamsters: Stitching Platforms & Audiences into Local Crisis Infrastructure. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 1277–1289, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4335-0. doi: 10.1145/2998181.2998290. URL http://doi.acm.org/10.1145/2998181.2998290. event-place: Portland, Oregon, USA.

[50] Mazen Danaf, Felix Becker, Xiang Song, Bilge Atasoy, and Moshe Ben-Akiva. Online discrete choice models: Applications in personalized recommendations. *Decision Support Systems*, 119:35–45, April 2019. ISSN 0167-9236. doi: 10.1016/j.dss.2019.02.003. URL https://www.sciencedirect.com/science/article/pii/S0167923619300181.

[51] Mark S. Daskin and Kayse Lee Maass. The p-Median Problem. In Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama, editors, *Location Science*, pages 21–45. Springer International Publishing, Cham, 2015. ISBN 978-3-319-13111-5. doi: 10.1007/978-3-319-13111-5_2.

[52] Dingxiong Deng, Cyrus Shahabi, and Ugur Demiryurek. Maximizing the Number of Worker's Self-Selected Tasks in Spatial Crowdsourcing. In *SIGSPATIAL'13: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 10, Orlando, Florida, USA, November 2013.

[53] Department of Homeland Security. Election Security, March 2018. URL https://www.dhs.gov/topic/election-security.

[54] Bishnu Prasad Devkota, Brent Doberstein, and Sanjay K. Nepal. Social Capital and Natural Disaster: Local Responses to 2015 Earthquake in Kathmandu. *International Journal of Mass Emergencies & Disasters*, 34(3):439–466, November 2016. ISSN 02807270. URL http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=sih&AN=123323447&site=ehost-live&scope=site.

[55] A. Dirr and M. Spicuzza. What we know so far about why Milwaukee only had 5 voting sites for Tuesday's election while Madison had 66. Milwaukee Journal Sentinel, April 10, 2020.

[56] Anthony Downs. *An economic theory of democracy*. New York : Harper and Brothers, 1957. URL http://archive.org/details/economictheoryof00down.

[57] Shawn Doyle. A Graph Partitioning Model of Congressional Redistricting. *Rose-Hulman Undergraduate Mathematics Journal*, 16(2), January 2017. URL https://scholar.rose-hulman.edu/rhumj/vol16/iss2/3.

[58] Thomas E Drabek. *Human system responses to disaster: An inventory of sociological findings*. Springer Science & Business Media, 1986.

[59] C. W Duin, A Volgenant, and S Voß. Solving group Steiner problems as Steiner problems. *European Journal of Operational Research*, 154(1):323–329, April 2004. ISSN 0377-2217. doi: 10.1016/S0377-2217(02)00707-5. URL https://www.sciencedirect.com/science/article/pii/S0377221702007075.

[60] Guillermo Durán, Mauro Giormenti, Mario Guajardo, Pablo M. Pinto, Pablo A. Rey, and Nicolás E. Stier-Moses. Improving access to voting with optimized matchings. *Electoral Studies*, 51:38–48, February 2018. ISSN 0261-3794. doi: 10.1016/j.electstud.2017.11.004.

[61] Joshua J. Dyck and James G. Gimpel. Distance, Turnout, and the Convenience of Voting*. *Social Science Quarterly*, 86(3):531–548, 2005. ISSN 1540-6237. doi: 10.1111/j.0038-4941.2005.00316.x. URL http://onlinelibrary.wiley.com/doi/abs/10.1111/j.0038-4941.2005.00316.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0038-4941.2005.00316.x.

[62] Henry L. Díaz, Karin A. Imitola, and Rolando J. A. Amado. Or/ms research perspectives in disaster operations management: a literature review. *Revista Facultad de Ingeniería Universidad de Antioquia*, (91):43–59, Apr 2019.

[63] EAC Advisory Board and Standards Board. Elections Operations Assessment: Threat Trees and Matrices and Threat Instance Risk Analyzer (TIRA). Technical report, University of South Alabama, December 2009.

[64] William A Edelstein. New voting systems for ny—long lines and high cost. *New Yorkers for Verified Voting*, 10, 2006.

[65] William A Edelstein and Arthur D Edelstein. Queuing and elections: Long lines, dres and paper ballots. In *EVT/WOTE*, 2010.

[66] Tony Evers, Joel Brennan, and Dawn Vick. Demographic Services Center's 2019 Population Estimates: Wisconsin's Moderate Growth Continues. Technical report, State of Wisconsin Department of Administration, 2019.

[67] Mauro Falasca and Christopher Zobel. An optimization model for volunteer assignments in humanitarian organizations. *Socio-Economic Planning Sciences*, 46(4): 250–260, December 2012. ISSN 0038-0121. doi: 10.1016/j.seps.2012.07.003. URL http://www.sciencedirect.com/science/article/pii/S0038012112000353.

[68] Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseininia, and Mark Goh. Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407, February 2012. ISSN 0360-8352. doi: 10.1016/j.cie.2011.08.020. URL https://www.sciencedirect.com/science/article/pii/S0360835211002490X.

[69] Lauren S. Fernandez. *Volunteer management system design and analysis for disaster response and recovery*. D.Sc., The George Washington University, United States – District of Columbia, 2007. URL https://search.proquest.com/docview/304875080/abstract/C85D64548A24622PQ/1.

[70] Benjamin Fifield, Michael Higgins, Kosuke Imai, and Alexander Tarr. Automated Redistricting Simulation Using Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 29(4):715–728, October 2020. ISSN 1061-8600, 1537-2715. doi: 10.1080/10618600.2020.1739532. URL https://www.tandfonline.com/doi/full/10.1080/10618600.2020.1739532.

[71] Matteo Fischetti, Markus Leitner, Ivana Ljubic, Martin Luipersbeck, Michele Monaci, Max Resch, Domenico Salvagnin, and Markus Sinnl. Thinning out Steiner trees: A node based model for uniform edge costs. *Mathematical Programming Computation*, 9, September 2016. doi: 10.1007/s12532-016-0111-0.

[72] Zack Fitzsimmons and Omer Lev. Selecting Voting Locations for Fun and Profit. *arXiv:2003.06879 [cs]*, March 2020. URL http://arxiv.org/abs/2003.06879. arXiv: 2003.06879.

[73] Charles Franklin. Marquette Law School Poll – May 3-7, 2020, May 2020. URL https://law.marquette.edu/poll/2020/05/12/detailed-results-of-the-marquette-law-school-poll-may-3-7-2020. Accessed on Aug. 5, 2020.

[74] Thomas Fujiwara, Kyle Meng, and Tom Vogl. Habit Formation in Voting: Evidence from Rainy Elections. *American Economic Journal: Applied Economics*, 8(4):160–188, October 2016. ISSN 1945-7782. doi: 10.1257/app.20140533. URL https://www.aeaweb.org/articles?id=10.1257/app.20140533.

[75] Gina Galindo and Rajan Batta. Review of recent developments in OR/MS research in disaster operations management. *European Journal of Operational Research*, 230 (2):201–211, October 2013. ISSN 0377-2217. doi: 10.1016/j.ejor.2013.01.039. URL http://www.sciencedirect.com/science/article/pii/S0377221713000866.

[76] Michael R. Garey. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco : W. H. Freeman, 1979. ISBN 978-0-7167-1045-5.

[77] Michel Gendreau, Alain Hertz, and Gilbert Laporte. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. *Operations Research*, 40 (6):1086–1094, 1992. ISSN 0030-364X. URL http://www.jstor.org/stable/171722. Publisher: INFORMS.

[78] Michel Gendreau, Gilbert Laporte, and Frédéric Semet. The covering tour problem. *Operations Research*, 45(4):568–576, 1997. URL https://EconPapers.repec.org/RePEc:inm:oropre:v:45:y:1997:i:4:p:568-576.

[79] Gianpaolo Ghiani, Francesca Guerriero, and Roberto Musmanno. The capacitated plant location problem with multiple facilities in the same site. *Computers & Operations Research*, 29(13):1903–1912, November 2002. ISSN 0305-0548. doi: 10.1016/S0305-0548(01)00065-X.

[80] J. G. Gimpel and J. E. Schuknecht. Political participation and the accessibility of the ballot box. *Political Geography*, 5(22):471–488, 2003. ISSN 0962-6298. doi: 10.1016/S0962-6298(03)00029-5.

[81] Garrett Glasgow and R. Michael Alvarez. Discrete Choice Methods. Technical report, August 2008. URL https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199286546.001.0001/oxfordhb-9780199286546-e-22. ISBN: 9780199286546.

[82] Lynn Gordon and Erhan Erkut. Improving Volunteer Scheduling for the Edmonton Folk Festival. *INFORMS Journal on Applied Analytics*, 34(5):367–376, October 2004. ISSN 0092-2102. doi: 10.1287/inte.1040.0097. URL http://pubsonline.informs.org/doi/abs/10.1287/inte.1040.0097.

[83] Sean Greene and Kyle Ueyama. Vote-By-Mail Rates More Than Double Since 2000, April 2015. URL https://www.pewtrusts.org/en/research-and-analysis/blogs/stateline/2015/4/29/vote-by-mail-practices-more-than-double-since-2000. Accessed on Mar. 16, 2021.

[84] Andrey Gubichev and Thomas Neumann. Fast approximation of steiner trees in large graphs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 1497–1501, New York, NY, USA, October 2012. Association for Computing Machinery. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398460. URL http://doi.org/10.1145/2396761.2398460.

[85] Gurobi Optimization. Python examples. URL https://www.gurobi.com/documentation/9.1/examples/tsp_py.html. Accessed on Feb. 21, 2021.

[86] S. L. Hakimi. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, 12(3):450–459, June 1964. ISSN 0030-364X. doi: 10.1287/opre.12.3.450.

[87] Hanna Halaburda, Mikołaj Jan Piskorski, and Pınar Yıldırım. Competing by Restricting Choice: The Case of Matching Platforms. *Management Science*, 64(8): 3574–3594, July 2017. ISSN 0025-1909. doi: 10.1287/mnsc.2017.2797. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2017.2797. Publisher: INFORMS.

[88] Moshe Haspel and H. Gibbs Knotts. Location, Location, Location: Precinct Placement and the Costs of Voting. *The Journal of Politics*, 67(2):560–573, May 2005. ISSN 0022-3816. doi: 10.1111/j.1468-2508.2005.00329.x. URL http://www.journals.uchicago.edu/doi/full/10.1111/j.1468-2508.2005.00329.x. Publisher: The University of Chicago Press.

[89] Umair Ul Hassan and Edward Curry. A Multi-armed Bandit Approach to Online Spatial Task Assignment. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pages 212–219, December 2014. doi: 10.1109/UIC-ATC-ScalCom.2014.68.

[90] Naieme Hazrati, Mehdi Elahi, and Francesco Ricci. Analysing Recommender Systems Impact on Users' Choices. In *ImpactRS Workshop*, page 3, Copenhagen, Denmark, September 2019.

[91] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan Political Redistricting by Computer. *Operations Research*, 13(6):998–1006, December 1965. ISSN 0030-364X. doi: 10.1287/opre.13.6.998.

[92] Benjamin Highton. Long Lines, Voting Machine Availability, and Turnout: The Case of Franklin County, Ohio in the 2004 Presidential Election. *PS: Political Science and Politics*, 39(1):65–68, 2006. ISSN 1049-0965. URL http://www.jstor.org/stable/20451680. Publisher: [American Political Science Association, Cambridge University Press].

[93] Maria Camila Hoyos, Ridley S. Morales, and Raha Akhavan-Tabatabaei. OR models with stochastic components in disaster operations management: A literature survey. *Computers & Industrial Engineering*, 82:183–197, April 2015. ISSN 0360-8352. doi: 10.1016/j.cie.2014.11.025. URL http://www.sciencedirect.com/science/article/pii/S0360835214004136.

[94] Axel Huord, Lane Baker, Alexandra Popke, Garrett Jensen, and Gabriella Garcia. Where Can You Drop Off Your Ballot? A 50 State Analysis. Technical report, HealthyElections.org, October 2020. URL https://healthyelections.org/sites/default/files/2020-10/Ballot_Drop_Off_0.pdf.

[95] International Medical Corps. Hurricane Dorian Situation Report #17. Technical report, December 2019. URL https://reliefweb.int/sites/reliefweb.int/files/resources/Int%20Med%20Corps%20-%20Hurricane%20Dorian_SitRep%2317.pdf.

[96] Kazuo Iwama and Shuichi Miyazaki. A Survey of the Stable Marriage Problem and Its Variants. In *International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*, pages 131–136, Kyoto, Japan, 2008. IEEE. ISBN 978-0-7695-3128-1. doi: 10.1109/ICKS.2008.7. URL http://ieeexplore.ieee.org/document/4460480/.

[97] Hai Jiang, Xin Qi, and He Sun. Choice-Based Recommender Systems: A Unified Approach to Achieving Relevancy and Diversity. *Operations Research*, 62(5):973–993, July 2014. ISSN 0030-364X. doi: 10.1287/opre.2014.1292. URL http://pubsonline.informs.org/doi/abs/10.1287/opre.2014.1292. Publisher: INFORMS.

[98] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately Interpreting Clickthrough Data as Implicit Feedback. *ACM SIGIR Forum*, August 2017. URL https://dl.acm.org/doi/abs/10.1145/3130332.3130334.

[99] Joint COVID Working Group. *Ballot Drop Box*. Cybersecurity and Infrastructure Security Agency (CISA) Elections Infrastructure Government Coordinating Council and Sector Coordinating Council, 245 Murray Lane Washington, D.C. 20528-0380, 1 edition, 7 2020. https://static1.squarespace.com/static/5a665c98017db2b60bc22084/t/5e8f42d717ee5e7ee2db8c8b/1586447064805/Ballot_Drop-Box_final.pdf.

[100] B. Kalyanasundaram and K. Pruhs. Online Weighted Matching. *Journal of Algorithms*, 14(3):478–488, 1993.

[101] Kaspari, Matthew. *Optimal Volunteer Assignment with an Application to the Denver B-Cycle Bike Sharing Program*. PhD thesis, University of Colorado Denver, 2005.

[102] Marc-André Kaufhold and Christian Reuter. The Self-Organization of Digital Volunteers across Social Media: The Case of the 2013 European Floods in Germany. *Journal of Homeland Security and Emergency Management*, 13(1), January 2016. ISSN 1547-7355, 2194-6361. doi: 10.1515/jhsem-2015-0063. URL https://www.degruyter.com/view/j/jhsem.2016.13.issue-1/jhsem-2015-0063/jhsem-2015-0063.xml.

[103] Leyla Kazemi and Cyrus Shahabi. GeoCrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 189–198, Redondo Beach, California, November 2012. Association for Computing Machinery. ISBN 978-1-4503-1691-0. doi: 10.1145/2424321.2424346. URL http://doi.org/10.1145/2424321.2424346.

[104] Kamyoung Kim. A Spatial Optimization Approach for Simultaneously Districting Precincts and Locating Polling Places. *ISPRS International Journal of Geo-Information*, 9(5):301, May 2020. ISSN 2220-9964. doi: 10.3390/ijgi9050301.

[105] Catherine King and Lawrence M. Leemis. Data analysis and simulation: Optimizing voter wait times. In *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, pages 199–204, April 2016. doi: 10.1109/SIEDS.2016.7489298.

[106] Marina Kogan, Leysia Palen, and Kenneth M. Anderson. Think Local, Retweet Global: Retweeting by the Geographically-Vulnerable During Hurricane Sandy. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pages 981–993, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2922-4. doi: 10.1145/2675133.2675218. URL http://doi.acm.org/10.1145/2675133.2675218. event-place: Vancouver, BC, Canada.

[107] A Gürhan Kök, Marshall L Fisher, and Ramnath Vaidyanathan. Assortment planning: Review of literature and industry practice. *Retail supply chain management*, pages 175–236, 2015.

[108] Anand Kulkarni, Philipp Gutheim, Prayag Narula, David Rolnitzky, Tapan Parikh, and Björn Hartmann. MobileWorks: Designing for Quality in a Managed Crowdsourcing Architecture. *IEEE Internet Computing*, 16(5):28–35, September 2012. ISSN 1941-0131. doi: 10.1109/MIC.2012.72.

[109] Richard C Larson. OR forum—perspectives on queues: Social justice and the psychology of queueing. *Operations research*, 35(6):895–905, 1987.

[110] Kyle Lassiter, Amin Khademi, and Kevin M. Taaffe. A robust optimization approach to volunteer management in humanitarian crises. *International Journal of Production Economics*, 163:97–111, May 2015. ISSN 0925-5273. doi: 10.1016/j.ijpe.2015.02.018. URL http://www.sciencedirect.com/science/article/pii/S0925527315000523.

[111] Harry A. Levin and Sorelle A. Friedler. Automated Congressional Redistricting. *ACM Journal of Experimental Algorithmics*, 24:1.10:1–1.10:24, April 2019. ISSN 1084-6654. doi: 10.1145/3316513. URL http://doi.org/10.1145/3316513.

[112] Jingsheng Li, Theodore T Allen, and Kimiebi Akah. Could simulation optimization have prevented 2012 central Florida election lines? In *2013 Winter Simulations Conference (WSC)*, pages 2088–2096. IEEE, 2013.

[113] Irene Lo, Vahideh Manshadi, Scott Rodilitz, and Ali Shameli. Commitment on Volunteer Crowdsourcing Platforms: Implications for Growth and Engagement. *arXiv*, September 2020.

[114] Thomas Ludwig, Christoph Kotthaus, Christian Reuter, Sören van Dongen, and Volkmar Pipek. Situated crowdsourcing during disasters: Managing the tasks of spontaneous volunteers through public displays. *International Journal of Human-Computer Studies*, 102:103–121, June 2017. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2016.09.008. URL http://www.sciencedirect.com/science/article/pii/S1071581916301197.

[115] Lohithaksha M. Maiyar and Jitesh J. Thakkar. Modelling and analysis of intermodal food grain transportation under hub disruption towards sustainability. *International Journal of Production Economics*, 217:281–297, November 2019. ISSN 0925-5273. doi: 10.1016/j.ijpe.2018.07.021. URL https://www.sciencedirect.com/science/article/pii/S092552731830286X.

[116] Vahideh Manshadi and Scott Rodilitz. Online Policies for Efficient Volunteer Crowdsourcing. *arXiv*, September 2020.

[117] Antonia Mantonakis, Pauline Rodero, Isabelle Lesschaeve, and Reid Hastie. Order in choice: effects of serial position on preferences. *Psychological Science*, 20(11):1309–1312, November 2009. ISSN 1467-9280. doi: 10.1111/j.1467-9280.2009.02453.x.

[118] Vladimir Marianov and Daniel Serra. Location–Allocation of Multiple-Server Service Centers with Constrained Queues or Waiting Times. *Annals of Operations Research*, 111(1):35–50, March 2002. ISSN 1572-9338. doi: 10.1023/A:1020989316737.

[119] Lavanya Marla, Kaushik Krishnan, and Sarang Deo. Managing EMS systems with user abandonment in emerging economies. *IISE Transactions*, 0(0):1–18, July 2020. ISSN 2472-5854. doi: 10.1080/24725854.2020.1802086. URL https://doi.org/10.1080/24725854.2020.1802086.

[120] George Mavrotas. Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 213(2):455–465, July 2009. ISSN 0096-3003. doi: 10.1016/j.amc.2009.03.037. URL https://www.sciencedirect.com/science/article/pii/S0096300309002574.

[121] Maria E. Mayorga, Emmett J. Lodree, and Justin Wolczynski. The optimal assignment of spontaneous volunteers. *Journal of the Operational Research Society*, 68(9):1106–1116, September 2017. ISSN 0160-5682. doi: 10.1057/s41274-017-0219-2. URL https://orsociety.tandfonline.com/doi/abs/10.1057/s41274-017-0219-2.

[122] William McGuire, Benjamin Gonzalez O'Brien, Katherine Baird, Benjamin Corbett, and Loren Collingwood. Does Distance Matter? Evaluating the Impact of Drop Boxes on Voter Turnout. *Social Science Quarterly*, 101(5):1789–1809, September 2020. ISSN 1540-6237. doi: 10.1111/ssqu.12853. URL http://onlinelibrary.wiley.com/doi/10.1111/ssqu.12853. Publisher: John Wiley & Sons, Ltd.

[123] John E. McNulty, Conor M. Dowling, and Margaret H. Ariotti. Driving Saints to Sin: How Increasing the Difficulty of Voting Dissuades Even the Most Motivated Voters. *Political Analysis*, 17(4):435–455, 2009. ISSN 1047-1987.

[124] Milwaukee County. Census Blocks, Milwaukee County. Technical report, 2018. URL https://geodata.wisc.edu/catalog/MilwaukeeCounty_cd0302d0aba844d89a22f162e753baaf_3.

[125] Milwaukee Election Commission. Absentee Ballot Drop Boxes Available at 15 Locations across the City, Including 13 Milwaukee Public, September 2020. URL https://city.milwaukee.gov/mayorbarrett/News/2020-News/Absentee-Ballot-Drop-Boxes-Available-at-15-Locations-across-the-City-Including-13-Milwaukee-Public.

[126] Milwaukee Elections Commission. November 3, 2020 - general election. Nov. 6, 2020, Accessed on June 5, 2021, 2020.

[127] Milwaukee OpenData. Voting wards 2020. Accessed on Aug. 6, 2020. URL https://data.milwaukee.gov/dataset/voting-wards/resource/01139d6b-b65a-4d63-89da-b87f3986ff0d?inner_span=True.

[128] Milwaukee OpenData. 2016 Nov 8, Voter Turnout By Ward, 3 2018. URL https://data.milwaukee.gov/dataset/voter-turnout-by-ward-november-8-2016. Accessed on Aug. 6, 2020.

[129] Milwaukee OpenData. 2020 Spring Election Results - Voter Turnout by Ward, 2020. URL https://data.milwaukee.gov/dataset/april-7-2020-spring-election/resource/3e78f0b1-c195-483d-99ac-22158369c72f. Accessed on Aug. 6, 2020.

[130] MIT Election Data + Science Lab. Elections Performance Index, 2018. URL https://elections.mit.edu. Accessed on Aug. 6, 2020.

[131] MIT Election Data + Science Lab. Voting by mail and absentee voting, March 2021. URL https://electionlab.mit.edu/research/voting-mail-and-absentee-voting.

[132] MIT Election Data & Science Lab. Elections Performance Index, 2022. URL https://elections.mit.edu/#/data/.

[133] Seyed Shahab Mofidi and Jennifer A. Pazour. When is it beneficial to provide freelance suppliers with choice? A hierarchical approach for peer-to-peer logistics platforms. *Transportation Research Part B: Methodological*, 126:1–23, August 2019. ISSN 01912615. doi: 10.1016/j.trb.2019.05.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261518310117.

[134] K. Morris and P. Miller. Voting in a pandemic: COVID-19 and primary turnout in Milwaukee, Wisconsin. Technical report, Brennan Center for Justice at New York University School of Law, New York, NY, 2020.

[135] Kevin Morris and Peter Miller. Voting in a Pandemic: COVID-19 and Primary Turnout in Milwaukee, Wisconsin. *Urban Affairs Review*, 58(2):597–613, March 2022. ISSN 1078-0874. doi: 10.1177/10780874211005016.

[136] Hitomi Murakami, Toru Takemoto, and Koichi Sakamoto. Study on Search and Rescue Operations in the 1995 Hanshin-Awaji Earthquake– Analysis of Labor Work in Relation with Building Types. *Proceedings of the 12th World Conference on Earthquake Engineering*, 272:1–8, 2000.

[137] Keisuke Murakami. A generalized model and a heuristic algorithm for the large-scale covering tour problem. *RAIRO Oper. Res.*, 2018. doi: 10.1051/ro/2017090.

[138] Tadahiko Murata and Kenta Konishi. Making a Practical Policy Proposal for Polling Place Assignment Using Voting Simulation Tool. *SICE Journal of Control, Measurement, and System Integration*, 6(2):124–130, March 2013. ISSN 1882-4889. doi: 10.9746/jcmsi.6.124. URL `https://doi.org/10.9746/jcmsi.6.124`. Publisher: Taylor & Francis _eprint: https://doi.org/10.9746/jcmsi.6.124.

[139] Lakshmi Narke and Azra Nasreen. A Comprehensive Review of Approaches and Challenges of a Recommendation System. *International Journal of Research in Engineering, Science and Management*, 3(4):4, April 2020.

[140] National Conference of State Legislatures. Voting Outside the Polling Place: Absentee, All-Mail and other Voting at Home Options, March 2022. URL `https://www.ncsl.org/research/elections-and-campaigns/absentee-and-early-voting.aspx`. Accessed on Jun. 9, 2022.

[141] Eric K. Noji. *The Public Health Consequences of Disasters*. Oxford University Press, November 1996. ISBN 978-0-19-974768-9.

[142] N. Oberijé. Civil Response After Disasters the Use of Civil Engagement in Disaster Abatement. Technical report, Netherlands Institute for Safety, The Netherlands, 2007.

[143] Johannes Oehrlein and Jan-Henrik Haunert. A cutting-plane method for contiguity-constrained spatial aggregation. *Journal of Spatial Information Science*, 2017, December 2017. doi: 10.5311/JOSIS.2017.15.379.

[144] Richard Oloruntoba. A wave of destruction and the waves of relief: issues, challenges and strategies. *Disaster Prevention and Management: An International Journal*, 14(4): 506–521, January 2005. ISSN 0965-3562. doi: 10.1108/09653560510618348. URL `https://doi.org/10.1108/09653560510618348`.

[145] Claudia Paciarotti, Angela Cesaroni, and Maurizio Bevilacqua. The management of spontaneous volunteers: A successful model from a flood emergency in Italy. *International Journal of Disaster Risk Reduction*, 31:260–274, October 2018. ISSN 2212-4209. doi: 10.1016/j.ijdrr.2018.05.013. URL http://www.sciencedirect.com/science/article/pii/S2212420918302115. Publisher: Elsevier.

[146] Leysia Palen and Sophia B Liu. Citizen Communications in Crisis: Anticipating a Future of ICT-Supported Public Participation. In *CHI Emergency Action*, page 10, 2007.

[147] Heather Paradis, Julie Katrichis, Michael Stevenson, Nicholas Tomaro, Rachel Mukai, Griselle Torres, Sanjib Bhattacharyya, Jeanette Kowalik, Karen Schlanger, and Eva Leidman. Public Health Efforts to Mitigate COVID-19 Transmission During the April 7, 2020, Election — City of Milwaukee, Wisconsin, March 13–May 5, 2020. Technical Report Vol. 69, No. 30, US Department of Health and Human Services/Centers for Disease Control and Prevention, 1600 Clifton Rd, Atlanta, GA 30333, July 2020.

[148] Kyle E. Paret, Maria E. Mayorga, and Emmett J. Lodree. Assigning spontaneous volunteers to relief efforts under uncertainty in task demand and volunteer availability. *Omega*, 99:102228, March 2021. ISSN 0305-0483. doi: 10.1016/j.omega.2020.102228. URL http://www.sciencedirect.com/science/article/pii/S0305048319300611.

[149] Nathaniel Persily. The American Voting Experience: Report and Recommendations of the Presidential Commission on Election Administration. Technical report, Presidential Commission on Election Administration, January 2014.

[150] Nathaniel Persily, Robert F. Bauer, Benjamin L. Ginsberg, Brian Britton, Joe Echevarria, Trey Grayson, Larry Lomax, Michele Coleman Mayes, Ann McGeehan, Tammy Patrick, and Christopher Thomas. The American Voting Experience: Report and Recommendations of the Presidential Commission on Election Administration. Technical report, Presidential Commission on Election Administration, January 2014. URL https://law.stanford.edu/publications/the-american-voting-experience-report-and-recommendations-of-the-presidential-commission-on-election-administration/.

[151] Stephen Pettigrew. The Racial Gap in Wait Times: Why Minority Precincts Are Underserved by Local Election Officials: RACIAL GAP IN WAIT TIMES. *Political Science Quarterly*, 132(3):527–547, September 2017. ISSN 00323195. doi: 10.1002/polq.12657.

[152] Pew Research Center. Sharp Divisions on Vote Counts, as Biden Gets High Marks for His Post-Election Conduct. Technical report, November 2020.

[153] Pew Research Center. The 2020 voting experience: Coronavirus, mail concerns factored into deciding how to vote, November 2020. URL https://www.pewresearch.org/politics/2020/11/20/the-voting-experience-in-2020/.

[154] Jason Pohl. FEMA rescuers collected $92m, often for few rescues. URL https://www.azcentral.com/story/news/local/phoenix/2018/02/18/fema-rescuers-spent-more-money-traveling-than-rescues/894687001/.

[155] S. Powers and D. Damrom. Analysis: 201,000 in Florida didn't vote because of long lines. Orlando Sentinel, January 29, 2013, 2013.

[156] Vladimir P. Reshetin and James L. Regens. Simulation Modeling of Anthrax Spore Dispersion in a Bioterrorism Incident. *Risk Analysis*, 23(6):1135–1145, 2003. ISSN 1539-6924. doi: 10.1111/j.0272-4332.2003.00387.x. URL http://onlinelibrary.wiley.com/doi/abs/10.1111/j.0272-4332.2003.00387.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0272-4332.2003.00387.x.

[157] Christian Reuter and Marc-André Kaufhold. Fifteen years of social media in emergencies: A retrospective review and future directions for crisis Informatics. *Journal of Contingencies and Crisis Management*, 26(1):41–57, 2018. ISSN 1468-5973. doi: 10.1111/1468-5973.12196. URL http://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5973.12196.

[158] Federica Ricca, Andrea Scozzari, and Bruno Simeone. Political Districting: from classical models to recent approaches. *Annals of Operations Research*, 204(1):271–299, April 2013. ISSN 02545330. doi: http://dx.doi.org/10.1007/s10479-012-1267-2.

[159] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to Recommender Systems Handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_1. URL https://doi.org/10.1007/978-0-387-85820-3_1.

[160] Richland County Voter Registration and Elections Office. June 9, 2020 Statewide Primary- Precincts and Polling Location. Dataset, May 2020.

[161] William H. Riker and Peter C. Ordeshook. A Theory of the Calculus of Voting. *The American Political Science Review*, 62(1):25–42, 1968. ISSN 0003-0554. doi: 10.2307/1953324. URL https://www.jstor.org/stable/1953324.

[162] Scott E. Sampson. Optimization of volunteer labor assignments. *Journal of Operations Management*, 24(4):363–377, June 2006. ISSN 0272-6963. doi: 10.1016/j.jom.2005.05.005. URL http://www.sciencedirect.com/science/article/pii/S0272696305001191.

[163] Natalie M. Scala, Paul Goethals, Josh Dehlinger, Yeabsira Mezgebe, Betelhem Jilcha, and Isabella Bloomquist. Evaluating Mail-in Security for Electoral Processes Using Attack Trees. *Risk Analysis*, January 2022.

[164] Brittany Schaefer and Bay Gammans. RI Board of Elections hopes to install security cameras at mail ballot drop boxes, September 2020. URL https://www.wpri.com/news/elections/ri-board-of-elections-hopes-to-install-security-cameras-at-mail-ballot-drop-boxes/.

[165] Zachary Scherer. Majority of Voters Used Nontraditional Methods to Cast Ballots in 2020, April 2021. URL https://www.census.gov/library/stories/2021/04/what-methods-did-people-use-to-vote-in-2020-election.html. Section: Government.

[166] Adam Schmidt and Laura A. Albert. Designing pandemic-resilient voting systems. *Socio-Economic Planning Sciences*, page 101174, October 2021. ISSN 0038-0121. doi: 10.1016/j.seps.2021.101174.

[167] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, June 1998. ISBN 978-0-471-98232-6. Google-Books-ID: zEzW5mhppB8C.

[168] SCvotes.org. South Carolina Precinct Information, 2022. URL https://www.scvotes.gov/cgi-bin/scsec/precinctnew?countykey=RICHLAND. Accessed on 2022-01-14.

[169] Takeshi Shirabe. A Model of Contiguity for Spatial Unit Allocation. *Geographical Analysis*, 37(1):2–16, 2005. ISSN 1538-4632. doi: 10.1111/j.1538-4632.2005.00605.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1538-4632.2005.00605.x.

[170] Lee Sigelman and William D. Berry. Cost and the calculus of voting. *Political Behavior*, 4(4):419–428, December 1982. ISSN 1573-6687. doi: 10.1007/BF00986972. URL https://doi.org/10.1007/BF00986972.

[171] Dan Simmons. Wisconsin election officials go from famine to feast with a swell of poll workers. *Washington Post*, October 2020. ISSN 0190-8286. URL https://www.washingtonpost.com/politics/wisconsin-election-officials-go-from-famine-to-feast-with-a-swell-of-poll-workers/2020/10/08/e5ea4398-08c2-11eb-a166-dc429b380d10_story.html.

[172] Gloria Simo. Sustaining Cross-Sector Collaborations: Lessons from New Orleans. *Public Organization Review*, 9(4):367, November 2009. ISSN 1573-7098. doi: 10.1007/s11115-009-0091-x. URL https://doi.org/10.1007/s11115-009-0091-x.

[173] Gloria Simo and Angela L. Bies. The Role of Nonprofits in Disaster Response: An Expanded Model of Cross-Sector Collaboration. *Public Administration Review*, 67(s1): 125–142, December 2007. ISSN 0033-3352. doi: 10.1111/j.1540-6210.2007.00821.x. URL http://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6210.2007.00821.x.

[174] N. C. Simpson and P. G. Hancock. Fifty Years of Operational Research and Emergency Response. *The Journal of the Operational Research Society*, 60:s126–s139, 2009. ISSN 0160-5682. URL www.jstor.org/stable/40206731.

[175] Colin T.S. Sng and David F. Manlove. Popular matchings in the weighted capacitated house allocation problem. *Journal of Discrete Algorithms*, 8(2):102–116, June 2010. ISSN 15708667. doi: 10.1016/j.jda.2008.11.008. URL https://linkinghub.elsevier.com/retrieve/pii/S1570866709000719.

[176] Lawrence V. Snyder. Facility location under uncertainty: a review. *IIE Transactions*, 38(7):547–564, June 2006. ISSN 0740-817X. doi: 10.1080/07408170500216480.

[177] Elections Systems & Software. "Prepared in Mind and Resources" — More Than a State Motto, February 2020. URL https://www.essvote.com/blog/our-customers/ess-case-study-south-carolina/. Accessed 2022-03-28.

[178] South Carolina Election Commission. Richland - 2016 Statewide General Election Results, 2016. URL https://www.enr-scvotes.org/SC/Richland/64698/183645/en/vt_data.html. Accessed 2022-03-28.

[179] South Carolina Election Commission. South Carolina's Paper-Based Voting System, 2019. URL https://www.scvotes.gov/sites/default/files/scec_10856_01_Cast_Your_Ballot_8.5%20X%2011_2019_02.pdf.

[180] South Carolina Election Commission. Richland - 2020 Statewide Primaries Election Night Reporting, 2020. URL https://www.enr-scvotes.org/SC/Richland/103442/Web02.252502/#/index.html. Accessed 2022-03-28.

[181] Douglas M Spencer and Zachary S Markovits. Long lines at polling stations? observations from an election day field study. *Election Law Journal*, 9(1):3–17, 2010.

[182] M. Spicuzza and A. Dirr. Why did Milwaukee have just 5 polling places? Aldermen want answers. Milwaukee Journal Sentinel, April 10, 2020.

[183] Mary Spicuzza. 'A very sad situation for voters': Milwaukeeans brave wait times as long as 2 1/2 hours, top election official says, April 2020. URL https://www.jsonline.com/story/news/politics/elections/2020/04/07/wisconsin-election-milwaukee-voters-brave-long-wait-lines-polls/2962228001/. Accessed on Aug. 8, 2020.

[184] Elections Project Staff. Drop boxes are trusted ballot return options., October 2020. URL https://bipartisanpolicy.org/blog/official-ballot-drop-boxes-are-secure-and-reliable/.

[185] Robert M. Stein and Greg Vonnahme. Engaging the Unengaged Voter: Vote Centers and Voter Turnout. *The Journal of Politics*, 70(2):487–497, April 2008. ISSN 0022-3816. doi: 10.1017/S0022381608080456. URL https://www.journals.uchicago.edu/doi/full/10.1017/S0022381608080456. Publisher: The University of Chicago Press.

[186] Robert M. Stein, Christopher Mann, Charles Stewart, Zachary Birenbaum, Anson Fung, Jed Greenberg, Farhan Kawsar, Gayle Alberda, R. Michael Alvarez, Lonna Atkeson, Emily Beaulieu, Nathaniel A. Birkhead, Frederick J. Boehmke, Joshua Boston, Barry C. Burden, Francisco Cantu, Rachael Cobb, David Darmofal, Thomas C. Ellington, Terri Susan Fine, Charles J. Finocchiaro, Michael D. Gilbert, Victor Haynes, Brian Janssen, David Kimball, Charles Kromkowski, Elena Llaudet, Kenneth R. Mayer, Matthew R. Miles, David Miller, Lindsay Nielson, Yu Ouyang, Costas Panagopoulos, Andrew Reeves, Min Hee Seo, Haley Simmons, Corwin Smidt, Farrah M. Stone, Rachel VanSickle-Ward, Jennifer Nicoll Victor, Abby Wood, and Julie Wronski. Waiting to Vote in the 2016 Presidential Election: Evidence from a Multi-county Study. *Political Research Quarterly*, 73(2):439–453, June 2020. Publisher: SAGE Publications Inc.

[187] Charles Stewart. 2016 survey of the performance of american elections. In *2016 Survey of the Performance of American Elections*. Harvard Dataverse, 2017. doi: 10.7910/DVN/Y38VIQ/SXXGGV. URL https://doi.org/10.7910/DVN/Y38VIQ/SXXGGV.

[188] Charles Stewart and Stephen Ansolabehere. Waiting to Vote. *Election Law Journal: Rules, Politics, and Policy*, 14(1):47–53, March 2015. ISSN 1533-1296. doi: 10.1089/elj.2014.0292.

[189] Charles Stewart III. Managing Polling Place Resources.pdf. Technical report, December 2015. URL http://web.mit.edu/vtp/Managing%20Polling%20Place%20Resources.pdf.

[190] Charles Stewart III and Stephen Ansolabehere. Waiting to vote. *Election Law Journal*, 14(1):47–53, 2015.

[191] Aaron Strauss. Poll Worker and Machine Optimization. http://web.mit.edu/vtp/calc2.html.

[192] Attila Tasnadi. The political districting problem: A survey. *Society and Economy*, 33 (3):543–554, 2011.

[193] The New York Times. Wisconsin Primary Recap: Voters Forced to Choose Between Their Health and Their Civic Duty. URL https://www.nytimes.com/2020/04/07/us/politics/wisconsin-primary-election.html. Accessed on Aug. 6, 2020.

[194] Hien To, Gabriel Ghinita, and Cyrus Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of the VLDB Endowment*, 7 (10):919–930, June 2014. ISSN 2150-8097. doi: 10.14778/2732951.2732966. URL http://doi.org/10.14778/2732951.2732966.

[195] Hien To, Cyrus Shahabi, and Leyla Kazemi. A Server-Assigned Spatial Crowdsourcing Framework. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 1(1):2:1–2:28, July 2015. ISSN 2374-0353. doi: 10.1145/2729713. URL http://doi.org/10.1145/2729713.

[196] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. Spatial crowdsourcing: a survey. *The VLDB Journal*, 29(1):217–250, January 2020. ISSN 0949-877X. doi: 10.1007/s00778-019-00568-7. URL https://doi.org/10.1007/s00778-019-00568-7.

[197] Constantine Toregas, Ralph Swain, Charles ReVelle, and Lawrence Bergman. The Location of Emergency Service Facilities. *Operations Research*, 19(6):1363–1373, October 1971.

[198] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, University Printing House Shaftesbury Road Cambridge CB2 8BS United Kingdom, July 2009. ISBN 978-0-521-76655-5. Google-Books-ID: R59uqw5HaM4C.

[199] John Twigg and Irina Mosel. Emergent groups and spontaneous volunteers in urban disaster response. *Environment and Urbanization*, 29(2):443–458, October 2017. ISSN 0956-2478. doi: 10.1177/0956247817721413. URL https://doi.org/10.1177/0956247817721413.

[200] Umair ul Hassan and Edward Curry. Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning. *Expert Systems with Applications*, 58:36–56, October 2016. ISSN 09574174. doi: 10.1016/j.eswa.2016.03.022. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417416301014.

[201] United States Census Bureau. Work Destination Analysis by Census Tracts. URL https://onthemap.ces.census.gov/tot/?q=milwaukee&ds=us_bgrp.

[202] United States Census Bureau. QuickFacts Milwaukee city, Wisconsin. Accessed on Aug. 5, 2020, July 2019. URL https://www.census.gov/quickfacts/fact/table/milwaukeecitywisconsin/PST045219.

[203] United States Census Bureau. Means of transportation to work by selected characteristics. Technical report, 2019. URL https://data.census.gov/cedsci/table?g=0500000US55079%241400000&y=2019&tid=ACSST5Y2019.S0802&hidePreview=false.

[204] United States Census Bureau. Race for the population 18 years and over, 2020. URL https://data.census.gov/cedsci/table?q=&g=0500000US55079%241000000&tid=DECENNIALPL2020.P3&hidePreview=true.

[205] Hamidreza Validi, Austin Buchanan, and Eugene Lykhovyd. Imposing Contiguity Constraints in Political Districting Models. *Operations Research*, December 2021. ISSN 0030-364X. doi: 10.1287/opre.2021.2141.

[206] Jan van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 259–278. SIAM, 2020.

[207] M. H van Hoorn and H. C Tijms. Approximations for the waiting time distribution of the M/G/c queue. *Performance Evaluation*, 2(1):22–28, May 1982. ISSN 0166-5316. doi: 10.1016/0166-5316(82)90018-9.

[208] Ervin Varga. Recommender Systems. In Ervin Varga, editor, *Practical Data Science with Python 3: Synthesizing Actionable Insights from Data*, pages 317–339. Apress, Berkeley, CA, 2019. ISBN 978-1-4842-4859-1. doi: 10.1007/978-1-4842-4859-1_8. URL https://doi.org/10.1007/978-1-4842-4859-1_8.

[209] Leticia Vargas, Nicolas Jozefowiez, and Sandra Ulrich Ngueveu. A dynamic programming operator for tour location problems applied to the covering tour problem. *Journal of Heuristics*, 23(1):53–80, 2017. URL https://ideas.repec.org/a/spr/joheur/v23y2017i1d10.1007_s10732-017-9324-2.html. Publisher: Springer.

[210] Matt Vasilogambros. Lawmakers Push to Preserve Pandemic Voting Access, December 2020. URL https://www.govtech.com/elections/Lawmakers-Push-to-Preserve-Pandemic-Voting-Access.html.

[211] Matt Vasilogambros, Carrie Levine, and Pratheek Rebala. National Data Release Sheds Light On Past Polling Place Changes, September 2020. URL https://pew.org/3473mZ5. Accessed 2022-02-15.

[212] Phebe Vayanos, Duncan McElfresh, Yingxiao Ye, John Dickerson, and Eric Rice. Active Preference Elicitation via Adjustable Robust Optimization. *arXiv*, March 2020. URL http://arxiv.org/abs/2003.01899. arXiv: 2003.01899.

[213] Tricia Wachtendorf and James M. Kendra. Rebel Food ... Renegade Supplies : Convergence After The World Trade Center Attack. Preliminary Papers 318, Disaster Research Center, 2001.

[214] Xinfang Wang, Muer Yang, and Michael J Fry. Efficiency and equity tradeoffs in voting machine allocation problems. *Journal of the Operational Research Society*, 66(8): 1363–1369, 2015.

[215] Wisconsin Elections Commission. 2016 General Election - Summary Statement of the Board of Canvassers. Accessed on Jun. 5, 2021, 2016. URL https://county.milwaukee.gov/files/county/county-clerk/Election-Commission/ElectionResultsCopy-1/2016Copy-1/Fall-General-ElectionCopy-1/11-8-16General2016_SummaryofCanvassCopy-1.pdf.

[216] Wisconsin Elections Commission. Absentee Ballot Report. Accessed on Jun. 5, 2021, 2016. URL https://elections.wi.gov/node/4414.

[217] Wisconsin Elections Commission. Elections and Coronavirus COVID-19. Presentation, July 2020.

[218] Wisconsin Elections Commission. April 7, 2020 absentee voting report. May 15, 2020, Accessed on Jul. 14, 2020., 2020. URL https://elections.wi.gov/sites/elections.wi.gov/files/2020-05/April%202020%20Absentee%20Voting%20Report.pdf.

[219] Wisconsin Elections Commission. Absentee ballot report - November 3, 2020 general election. Accessed on June 5, 2021, 2020. URL https://elections.wi.gov/node/6862.

[220] Meagan Wolfe and Richard Rydecki. Absentee ballot drop box information. Memo to All Wisconsin Election Officials, 8 2020.

[221] Muer Yang, Michael J Fry, and W David Kelton. Are all voting queues created equal? In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 3140–3149. IEEE, 2009.

[222] Muer Yang, Theodore T Allen, Michael J Fry, and W David Kelton. The call for equity: simulation optimization models to minimize the range of waiting times. *IIE Transactions*, 45(7):781–795, 2013.

[223] Muer Yang, Michael J Fry, W David Kelton, and Theodore T Allen. Improving voting systems through service-operations management. *Production and Operations Management*, 23(7):1083–1097, 2014.

[224] Muer Yang, Xinfang Jocelyn Wang, and Nuo Xu. A robust voting machine allocation model to reduce extreme waiting. *Omega*, 57:230–237, 2015.

[225] Güvenç Şahin and Haldun Süral. A review of hierarchical facility location models. *Computers & Operations Research*, 34(8):2310–2331, August 2007. ISSN 0305-0548. doi: 10.1016/j.cor.2005.09.005.

Appendix 7

Proofs

Proof to Remark 3.1. We provide a counter example in which the optimal solution does not have contiguous districts. Let constraint sets (3.7)-(3.9) be redundant, so we do not need to consider that allocation of server resources. Consider the example depicted in Figure 7.1. There are four population districts, $I = \{A, B, C, D\}$, and three polling locations, $J = \{0, 1, 2\}$. Let the population p_i of each district be one. Let the standard polling location of the population districts be $j(A) = j(B) = 0$, $j(C) = 1$, and $j(D) = 2$. There is one polling location that cannot be active $\hat{J} = \{0\}$. We allow up to two polling locations to be active $(n = 2)$. The distance increases $d_{ij} - d_{ij(i)}$ that would result from assigning $i \in \{A, B\}$ to $j \in \{1, 2\}$ are provided in Figure 7.1. The distance increase of assigning C to 2 or D to 1 is greater than zero. Thus, an optimal solution has two active polling locations (C and D).

There are four feasible solutions in which two polling locations are active. In each, C is assigned to 1 and D is assigned to 2 due to constraint set (3.6). The first feasible solution is to assign both A and B to 1; the OFV of this assignment is $2 + \epsilon$. The second is to assign both A and B to 2; the OFV of this assignment is $2 + \epsilon$. The third is to assign A to 2 and B to 1; the OFV of this assignment is $2 + 2\epsilon$. Finally, the fourth is to assign A to 1 and B to 2; the OFV of this assignment is 2. Thus, the fourth assignment is optimal when $\epsilon > 0$.

However, B is the only neighbor of A, so the optimal solution does represent contiguous polling location districts.

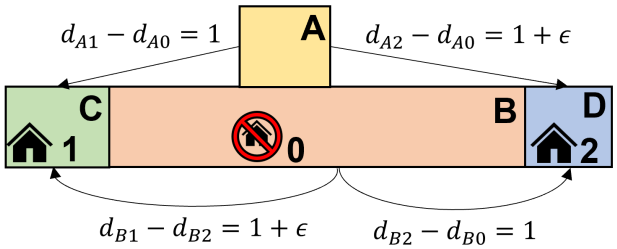


Figure 7.1: An example PLCP instance for which the optimal assignment is non-contiguous. Each rectangle represents a population district and each building represents a polling location.

□

Proof to Theorem 3.3. Given any solution $x \in \{0,1\}^{|I|\times|J|}$, $y \in \{0,1\}^{|J|\times c_j}$ feasible to constraint set (3.8), it must be feasible to the wait time cover. Suppose it is not. Then,

$$\sum_{i \in W_{jm}} x_{ij} > |W_{jm}| - 1 + y_{j(m+1)}$$

If $y_{j(m+1)} = 0$, then

$$\sum_{i \in W_{jm}} x_{ij} > |W_{jm}| - 1$$

which implies that

$$\sum_{i \in W_{jm}} \lambda_i x_{ij} > \lambda_{\alpha m}$$

by the definition of W_{jm}. This violates constraint set (3.8).

If $y_{j(m+1)} = 1$, then

$$\sum_{i \in W_{jm}} x_{ij} > |W_{jm}| - 1 + 1 = |W_{jm}|$$

which is a contradiction since each $x_{ij} \leq 1$. □

Proof to Theorem 3.4. Suppose we have a solution $x \in \{0,1\}^{|I|\times|J|}$, $y \in \{0,1\}^{|J|\times c_j}$ that satisfies constraints (3.2)-(3.11). The solution must be valid for the server-resource inequalities. Suppose it is not. Then, for some $i \in I$ and $j \in J$,

$$y_{jm_j^*} < x_{ij}$$

Since it is not possible that $y_{jm_j^*} < x_{ij}$ when $x_{ij} = 0$, we can assume that $x_{ij} = 1$ and $y_{jm_j^*} = 0$.

By constraint set (3.6), it must be true that $x_{kj} = 1$ for each $k \in I$ such that $j(k) = j$. By constraint set (3.10), we know that $y_{jm} = 0$ for every $m \geq m_j^*$. Thus,

$$\begin{aligned}
\sum_{k\in I:j(k)=j} \lambda_k &\leq \sum_{k\in I} \lambda_k x_{kj} \\
&\leq y_{j1}\lambda_{\alpha 1} + \sum_{m=2}^{c_j} y_{jm}(\lambda_{\alpha m} - \lambda_{\alpha(m-1)}) \\
&\leq \lambda_{\alpha 1} + \sum_{m=2}^{m_j^*-1} (\lambda_{\alpha m} - \lambda_{\alpha(m-1)}) \\
&\leq \lambda_{\alpha(m_j^*-1)}
\end{aligned} \tag{7.1}$$

This is a contradiction on the definition of m_j^*.

□

Proof to Theorem 3.5. The objective function (OFV) is non-negative for any feasible solution, since $p_i \geq 0$, $x_{ij} \geq 0$, and $\max\{0, d_{ij} - d_{ij(i)}\} \geq 0$ for each $i \in I$ and $j \in J$. If there exists a feasible assignment with an OFV of zero, it must be optimal.

The OFV of the solution defined by x^* is

$$\sum_{i \in I} \sum_{j \in J} p_i \max\{0, d_{ij} - d_{ij(i)}\} x^*_{ij} = \sum_{i \in I} p_i \max\{0, d_{ij(i)} - d_{ij(i)}\} x^*_{ij(i)} = 0$$

since $\max\{0, d_{ij(i)} - d_{ij(i)}\} = \max\{0, 0\} = 0$. Thus, it must be optimal.

□

Proof to Remark 3.6. We provide an example in which polling locations are consolidated in all feasible solutions. Suppose there are two population districts, $I = \{A, B\}$, each originally assigned to unique polling locations, $J = \{0, 1\}$. Let $j(A) = 0$ and $j(B) = 1$. Suppose $s_{\max} = 10$, $c_0 = c_1 = 10$, $n = 2$, and $R = \emptyset$. Suppose $\tau = 30$ minutes, $\mu_j = 1.005$ voters/minute, $\alpha = 0.95$, and $\lambda_A = \lambda_B = 5$.

We first show that the assignment $x_{A0} = 1$ and $x_{B1} = 1$ (and zero otherwise) does not satisfy the wait time constraints (3.8). Note that the value of $P(w > 30)$ is decreasing with an increasing number of server resources m distributed to a polling location. Thus, with $s_{\max} = 10$ we simply need to check whether the wait time constraints (3.8) are satisfied when 5 server resources are allocated to each polling location, otherwise the wait time constraint is violated for at least one polling location. The following is true:

$$P(w_0 > 30) = P(w_1 > 30) = \left(\frac{5\rho_5}{5 - 5/1.005}\right) e^{-30(5 \times 1.005 - 5)} \approx 0.094 > 0.05 = 1 - \alpha$$

Thus, the assignment of polling locations to their standard polling locations is not feasible when $s_{\max} = 10$.

However, suppose both population districts are assigned to the same polling location. Without loss of generality, let this be polling location 0. We then can distribute all 10 server resources to polling location 0 (since $10 \leq c_0$). The probability of waiting longer than 30

minutes is then:

$$P(w_0 > 30) = \left(\frac{10\rho_{10}}{10 - 10/1.005}\right) e^{-30(10\times 1.005 - 10)} \approx 0.022 < 0.05 = 1 - \alpha$$

Thus, this consolidated assignment is feasible for the wait time constraint.

□

Proof to Theorem 3.7. Suppose we have an instance to the p-median problem defined by the number of facilities to locate h (to avoid any confusion with the population parameter p_i in the PLCP), the demand nodes V, the facility locations L, the populations g_v of each $v \in V$, and the distances $f_{v\ell}$ for $v \in V$ and $\ell \in L$.

We can construct an instance to PLCP in polynomial time using the following sets and parameters. Let $I = V$ and $J = L \cup \{j_{\text{dummy}}\}$ where j_{dummy} is a "dummy" polling location. Let $\hat{J} = \{j_{\text{dummy}}\}$. Let $j(i) = j_{\text{dummy}}$ and let $i(j)$ be any arbitrary element of I for each $i \in I$ and $j \in J$. Let $p_v = g_v$ for each $v \in V = I$. Let $d_{v\ell} = f_{v\ell}$ for each $v \in V = I$ and $\ell \in L = J \setminus \{j_{\text{dummy}}\}$. Let $d_{ij(i)} = d_{ij_{\text{dummy}}} = \min_{v\in V, \ell \in L} f_{d\ell}$ for each $i \in I$. Let $n = h$, $R = \emptyset$, $c_j = 1$ for each $j \in J$, $s_{\max} = |J|$, $\lambda_i = 1$, $\lambda_{\alpha 1} = |I|$, and $N_i = I \setminus \{i\}$ for each $i \in I$.

We show that given a solution to the p-median problem, we can construct a feasible solution to the PLCP. Suppose we have a feasible solution to the p-median problem. Let $L' \subseteq L$ denote the h facilities selected in the solution to the p-median problem. Similarly, let $A' \subseteq \{(v, \ell) : v \in V,\ \ell \in L\}$ denote the assignment of demand to facilities. Then, we can construct a solution to the PLCP by letting

$$x_{ij} = \begin{cases} 1 & (i,j) \in A' \\ 0 & otherwise \end{cases} \qquad \forall\, i \in I, j \in J$$

and

$$y_{j1} = \begin{cases} 1 & j \in L' : \exists\, v \in V, (v,j) \in A' \\ 0 & otherwise \end{cases} \qquad j \in J$$

This is feasible to the PLCP. Constraint sets (3.4), (3.3), and (3.5) are trivially satisfied due to the feasibility of the p-median solution. Constraint sets (3.2) and (3.6) are satisfied, since $j_{\text{dummy}} \notin L'$. There are no constraints from constraint sets (3.7) or (3.10) instantiated, since $R = \emptyset$ and $c_j = 1$ for all $j \in J$, respectively. Constraint set (3.8) is equivalent to $\sum_{i \in I} x_{ij} \leq |I| y_{j1}$, and the satisfaction of this constraint is implied by constraint set (3.5). Constraint set (3.9) is trivially satisfied because there are at most $|J|$ open facilities and $c_j = 1$ for each $j \in J$. Constraint set (3.11) is satisfied, since there are no (i,j)-separators in a complete graph. Lastly, the binary constraints (3.12) and (3.13) are satisfied, since x and y take values of either one or zero.

We next show that given a feasible solution to the PLCP, $x \in \{0,1\}^{|I| \times |J|}$ and $y \in \{0,1\}^{|J| \times c_j}$, we can construct a feasible solution to the p-median problem. Let $L' = \{j : y_{j1} = 1\}$ be the selected facilities. This may have cardinally less than h. If so, arbitrarily add facilities to the solution that are not already selected so there are h facilities selected in total. Do not assign any population districts to these locations. Let $A' = \{(i,j) : x_{ij} = 1,\ i \in I,\ j \in J\}$ be the assignment of demands to facilities. This solution is trivially feasible to the p-median problem due to constraint sets (3.3), (3.4), and (3.5) of the PLCP.

Next, we argue that given an optimal solution to the PLCP instance, we can construct an optimal solution to the p-median problem. Let x^* and y^* define the optimal solution to the PLCP, and let L^* and A^* be the solution for the p-median problem obtained from this solution. Suppose that L^* and A^* do not define an optimal solution to the p-median problem. Then there must exist a feasible solution to the p-median problem, $\hat{L}$ and $\hat{A}$, such

that

$$\sum_{(v,\ell)\in\hat{A}} g_v f_{v\ell} < \sum_{(v,\ell)\in A^*} g_v f_{v\ell}$$

Let $\hat{x}$ and $\hat{y}$ be the solution for the PLCP constructed from $\hat{L}$ and $\hat{A}$ using the procedure previously outlined. The value of this solution is

$$\begin{aligned}\sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}\hat{x}_{ij} &= \sum_{i,j\in\hat{A}} p_i \max\{0, d_{ij} - d_{ij(i)}\} \\ &= \sum_{v,\ell\in\hat{A}} g_v(f_{v\ell} - \min_{v'\in V,\ell'\in L} f_{v'\ell'}) \\ &= \sum_{v,\ell\in\hat{A}} g_v f_{v\ell} - \sum_{v'\in V} g_{v'} \min_{v'\in V,\ell'\in L} f_{v'\ell'}\end{aligned}$$

The term $\sum_{v\in V} g_v \min_{v'\in V,\ell'\in L} f_{v'\ell'}$ is a constant. Thus, it must be true that

$$\sum_{v,\ell\in\hat{A}} g_v f_{v\ell} < \sum_{v,\ell\in A^*} g_v f_{v\ell}$$

$$\sum_{v,\ell\in\hat{A}} g_v f_{v\ell} - \sum_{v\in V} g_v \min_{v'\in V,\ell'\in L} f_{v'\ell'} < \sum_{v,\ell\in A^*} g_v f_{v\ell} - \sum_{v\in V} g_v \min_{v'\in V,\ell'\in L} f_{v'\ell'}$$

$$\sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}\hat{x}_{ij} < \sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}x^*_{ij}$$

which contradicts the assumption that $\hat{x}$ is an optimal solution to the PLCP.

Thus, if there exist a polynomial time algorithm that solves the PLCP, then the same algorithm can be used to solve the p-median problem in polynomial time. Since the p-median problem is NP-Hard [51], the PLCP is also NP-Hard.

□

Proof to Theorem 3.8. We provide a reduction of the bin packing problem to the PLCP. The bin packing problem is to identify whether a set of f objects of volume v_i can be assigned into q or fewer bins, each having a volume V.

Given an instance of the bin packing problem, we construct an instance to the PLCP. Let $I = \{1, \ldots, f\}$, $n = q$, $\lambda_i = v_i$, $\lambda_{\alpha 1} = V$. Also let $J = \{0, 1, 2, \ldots, q\}$ with $c_j = 1$ for each $j \in J$, $\hat{J} = \{0\}$, and $j(i) = 0$ for each $i \in I$. To instantiate the remainder of the parameters of the PLCP, we let $R = \emptyset$, $s_{\max} = \infty$, and $N_i = I \setminus \{i\}$ for each $i \in I$. With these parameter values, the feasible region of the PLCP reduces to

$$\sum_{i \in I} v_i x_{ij} \leq V y_{j1} \qquad \forall\, j \in J \setminus \hat{J} \tag{7.2}$$

$$\sum_{j \in J \setminus \hat{J}} y_{j1} \leq q \tag{7.3}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, i \in I, j \in J \setminus \hat{J} \tag{7.4}$$

$$y_{j1} \in \{0, 1\} \qquad \forall\, j \in J \setminus \hat{J} \tag{7.5}$$

The first constraint set (7.2) is a result of the wait time constraints (3.8) of the PLCP while the second constraint (7.3) is a result of constraint (3.3) of the PLCP.

Finding a solution that satisfies these constraints is equivalent to finding a solution to the bin packing problem. Thus, if there exists a polynomial time algorithm to determine if the feasible region of the PLCP is non-empty, the same algorithm can be used to solve the bin packing problem. Since the bin packing problem is NP-Complete [76], it is NP-Complete to determine if the PLCP has a feasible solution. □

Proof to Theorem 3.9. We begin by proving x^* is a feasible assignment if and only if the four conditions hold. We start by proving the "if" direction. Note that constraint set (3.4) of the PLCP is trivially satisfied by the construction of x^*. For x^* to be a feasible assignment, we must show that there exists a $y \in \{0, 1\}^{|J| \times c_j}$ such that x^*, y^* is feasible to the PLCP. So first, we must show that the following feasible region

$$\mathcal{F} := \{y \in \{0, 1\}^{|J| \times c_j} : (3.2), (3.3), (3.5), (3.6), (3.8) - (3.10), x_{ij} = x^*_{ij}\ \forall\, i \in I,\ j \in J\}$$

is non-empty. We argue the values $y_{jm} = 1$ for all $j \in J$ and $m \in \mathbb{Z}_+ : m \leq \hat{m}_j$ and zero otherwise belong to $\mathcal{F}$. We consider each constraint of $\mathcal{F}$ independently.

- For $i \in I$, $x^*_{ij} = 0$ for all $j \in \hat{J}$, since $z(i) \notin \hat{J}$. This implies $\sum_{i \in I: z(i)=j} \lambda_i = 0$ and $\hat{m}_j = 0$ for all $j \in \hat{J}$. Thus, $y_j = 0$ for all $j \in \hat{J}$ and constraint set (3.2) is satisfied.
- Constraint set (3.3) is satisfied by assumption of the second condition.
- By assumption within the PLCP, $\lambda_i > 0$ for each $i \in I$. Thus, constraint set (3.5) is satisfied by construction since $\hat{m}_j \geq 1$ for each $j \in J$ where $\sum_{i \in I: z(i)=j} \lambda_i > 0$.
- Constraint set (3.6) is satisfied since $q_i(j(i)) = 1$ by definition.
- Constraint set (3.8) is satisfied by construction of $\hat{m}_j$ and that $\hat{m}_j \leq c_j$ for each $j \in J$ by assumption of the third condition.
- The satisfaction of constraint set (3.9) is implied by the assumption of the third condition.
- Constraint set (3.10) is trivially satisfied by the construction of y.

We now show that x^*, y is feasible for the PLCP. The assumption of the fourth condition implies that x^* is feasible for constraint set (3.7). What remains to be checked is whether x^*, y is feasible to constraint set (3.11). We prove by contradiction: suppose the solution defined by x^* is not feasible for constraint set (3.11). Then there exists an $i \in I$ and (i, j)-separator C such thaconsolidation:T:

$$x^*_{iz(i)} > \sum_{k \in C} x^*_{kz(k)}$$

Since $x^*_{iz(i)} \in \{0, 1\}$, $\sum_{k \in C} x^*_{kz(k)} = 0$. By assumption of the first condition, there exists a path between i to $i(z(i))$ in $G[\{i' \in I : z(i') = z(i)\} \cup \{i(z(i))\}]$. Let P denote the nodes

visited on this path. By Definition 3.1, any (i, j)-separator C must contain at least one node belonging to P. These two statements imply thaconsolidation:T:

$$0 = \sum_{k \in C} x^*_{kz(k)} \geq \sum_{k \in C \cap P} x^*_{kz(k)} \geq 1$$

This is a contradiction.

We now prove the "only if" claim. We prove by contrapositive: if the four conditions do not hold, then x^* is not a feasible assignment. We consider each condition independently. We begin with the first condition. Suppose there does not exist a path within $G[\{i' \in I : z(i') = z(i)\} \cup \{i(z(i))\}]$ from i to $i(z(i))$ Let $C = \{i' \in I : z(i') \neq z(i)\}$. This is must be an (i, j)-separator and

$$1 = x_{iz(i)} > \sum_{k \in C} x_{kz(k)} = 0$$

which contradicts constraint set (3.11). We consider the fourth condition next. If $\sum_{i \in I} a_{rij} x^*_{ij} > \bar{a}_{rj}$ for some $j \in J$ and $r \in R$, then the assignment does not satisfy constraint set (3.7) for the PLCP.

For the remaining two conditions, we consider that if x^* is a feasible assignment, there must exist a $y \in \{0, 1\}^{|J| \times c_j}$ such that x^*, y is feasible to the PLCP. If no such y exists ($\mathcal{F} = \emptyset$), then x^* cannot be a feasible assignment. We consider the second statement. If $|\cup_{i \in I} z(i)| > n$, then constraint set (3.5) requires that $|\{j \in J : y_{j1} = 1\}| > n$, which contradicts constraint set (3.3). We now consider the third condition. Suppose $\hat{m}_j > c_j$ for some $j \in J$. Then $\sum_{i \in I} \lambda_i x_{ij} > \lambda_{\alpha c_j} \geq y_{j1}\lambda_{\alpha 1} + \sum_{m=2}^{c_j} y_{jm}(\lambda_{\alpha m} - \lambda_{\alpha(m-1)})$ for any $y \in \{0, 1\}^{|J| \times c_j}$, which contradicts constraint set (3.8). Suppose $\sum_{j \in J} \hat{m}_j > s_{\max}$. Then for any $y \in \{0, 1\}^{|J| \times c_j}$ such that $\sum_{j \in J} \sum_{m=1}^{c_j} y_{jm} \leq s_{\max}$, there must exist at least one $j \in J$ such that $\sum_{i \in I} \lambda_i x_{ij} > y_{j1}\lambda_{\alpha 1} + \sum_{m=2}^{c_j} y_{jm}(\lambda_{\alpha m} - \lambda_{\alpha(m-1)})$. This contradicts constraint set (3.8).

We now prove optimality. Suppose x^* is a feasible assignment. Then there exists a $y \in \{0, 1\}^{|J| \times c_j}$ such that x^*, y is feasible to the PLCP. The pair x^*, y is optimal to the PLCP.

Suppose it is not. Then there exists some alternate feasible assignment $\hat{x} \in \{0,1\}^{|I|\times|J|}$ such that

$$\sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}\hat{x}_{ij} < \sum_{i\in I}\sum_{j\in J} p_i \max\{0, d_{ij} - d_{ij(i)}\}\hat{x}_{ij} = \sum_{i\in I} p_i \max\{0, d_{i\hat{j}^i} - d_{ij(i)}\}$$

Let $\hat{z}(i)$ represent the assigned polling location for $i \in I$ in solution $\hat{x}$. Then, for at least one $i \in I$,

$$\max\{0, d_{i\hat{z}(i)} - d_{ij(i)}\} < \max\{0, d_{iz(i)} - d_{ij(i)}\}$$

If such a $\hat{z}(i)$ exists, then $q_i(\hat{z}(i)) < q_i(z(i))$, which contradicts the definition of $z(i)$. □

Proof to Theorem 4.1. We reduce the traveling salesman problem (TSP) to the DBLP. Suppose we have an instance of the symmetric TSP defined by the nodes $\bar{N}$, edges $\bar{E}$, and edge costs $\bar{c}$. We construct an instance of the DBLP as follows. Let $N = \bar{N}$, $T = \bar{N}$, $W = \emptyset$, $r = 0$, $q = 0$, $f_j = 0$ for each $j \in \bar{N}$, $E = \bar{E}$, and $c_{ij} = \bar{c}_{ij}$ for $(i,j) \in \bar{E}$. Then, the DBLP is equivalent to:

$$\min_{x,y} z_1 = \sum_{(i,j)\in\bar{E}} \bar{c}_{ij} x_{ij} \tag{7.6}$$

$$\text{s.t.} \sum_{i\in N:(i,j)\in\bar{E}} x_{ij} = 2 \qquad \forall\, j \in \bar{N} \tag{7.7}$$

$$\sum_{i\in S, j\in N\setminus S} x_{ij} \geq 2 \qquad \forall S \subset \bar{N},\ 2 \leq |S| \leq |\bar{N}| - 2 \tag{7.8}$$

$$x_{ij} \in \{0,1\} \qquad \forall\,(i,j) \in \bar{E} \tag{7.9}$$

The formulation follows from the fact that constraint sets (4.2) and (4.3) are empty since $W = \emptyset$, and constraint set (4.4) requires that y_j is to equal one for each $j \in T = \bar{N}$. This equivalent formulation is an instance of the TSP. Thus, if we can solve the DBLP in polynomial time, then we can solve the TSP in polynomial time. Since the TSP is NP-Hard,

so is the DBLP. □

Proof to Lemma 1. Suppose there was a set $N' \subseteq N$, such that $T \subseteq N'$, for which

$$\frac{v^1_w + \sum_{n \in N'} a_{nw}}{v^0_w + v^1_w + \sum_{n \in N'} a_{nw}} < \frac{v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}}{v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}}$$

Then it must be true that

$$\frac{v^0_w}{v^0_w + v^1_w + \sum_{n \in N'} a_{nw}} > \frac{v^0_{\hat{w}}}{v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}}$$

This implies that

$$\frac{v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}}}{v^0_w + v^1_w + \sum_{n \in N'} a_{nw}} > \frac{v^0_{\hat{w}}}{v^0_w} \geq 1$$

where the second inequality is true by the assumption of $v^0_{\hat{w}} \geq v^0_w$. This implies

$$v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in N'} a_{n\hat{w}} > v^0_w + v^1_w + \sum_{n \in N'} a_{nw}$$

$$\implies v^0_{\hat{w}} + v^1_{\hat{w}} + \sum_{n \in T} a_{n\hat{w}} + \sum_{n \in N' \setminus T} a_{n\hat{w}} > v^0_w + v^1_w + \sum_{n \in T} a_{nw} + \sum_{n \in N' \setminus T} a_{nw}$$

$$\implies (v^0_{\hat{w}} - v^0_w) + (v^1_{\hat{w}} + \sum_{n \in T} a_{n\hat{w}} - v^1_w - \sum_{n \in T} a_{nw}) + \sum_{n \in N' \setminus T} (a_{n\hat{w}} - a_{nw}) > 0$$

However, each parenthesis term is negative (or zero) by assumption, and the sum can never be greater than zero. This is a contradiction. □

Proof to Feasibility of DBLP Solution from Section 4.4.3. Let $\hat{x} \in \{0,1\}^{|E|}$ and $\hat{y} \in \{0,1\}^{|N|}$ be the feasible solution to the first CTP instance found. Let $\hat{T} := T \cup \{n \in N : \hat{y}_n = 1\}$ represent the updated set of required locations used when solving the second CTP instance. Let $x^* \in \{0,1\}^{|E|}$ and $y^* \in \{0,1\}^{|N|}$ be the feasible solution to the second CTP instance. Let $N^0 := \{n \in N : y^*_n = 1\}$ denote the drop box locations selected according to y^*. Given a valid solution procedure for the CTP, x^* describes a tour visiting N^0. Thus, the solution

must satisfy constraint sets (4.5) and (4.6) for the DBLP. Moreover, $T \subseteq \hat{T} \subseteq N^0$ for any feasible solution to the second CTP instance. Thus, the solution satisfies constraint set (4.4) for the DBLP. What remains to be verified is the satisfaction of constraint set (4.3). By construction:

$$\begin{aligned}
\sum_{n \in N_w} y_n^* &= |N_w \cap N^0| \\
&= |N_w \cap \hat{T}| + |N_w \cap (N^0 \setminus \hat{T})| \\
&\geq 1 + |N_w \cap (N^0 \setminus \hat{T})| \\
&\geq 1 + \sum_{n \in N_w \setminus \hat{T}} y_n^* \\
&\geq 1 + 1 \\
&\geq 2
\end{aligned}$$

The first equality follows from the definition of N^0. The second equality follow from the fact that $\hat{T} \subseteq N^0$. The third statement follows from the fact that $|N_w \cap \hat{T}| = |N_w \cap \{n \in N : \hat{y}_n = 1\}| \geq 1$. The fourth statement follows from the definition of N^0. The fifth statement follows from the fact that no location in $\hat{T}$ is a member of the covering sets in the second instance $(N_w \setminus \hat{T})$, and the second CTP must select another location to include within the tour to cover each $w \in W$.

□

Proof to Theorem 5.1. According to the Ghouila-Houri Theorem [167], it is sufficient to show that there is an assignment $s(R) : R \to \pm 1 : \sum_{r \in R} s(r)r$ has entries within $\{-1, 0, 1\}$ for every subset of rows, $R \subseteq A$ where A is the constraint matrix of the standard linear programming relaxation of the MOMM problem in the form $Ax \leq b$. We show that using Algorithm 3 and given any $R \subseteq A$, we can construct an assignment $s(R)$ such that $\sum_{r \in R} s(r)r$ has entries

within $\{-1, 0, 1\}$.

Let A' be the matrix A without the rows corresponding to constraints in constraint set (5.2). We first show Algorithm 3 succeeds in finding a suitable assignment $s(R)$ for $R \subseteq A'$. Subsequently, we show how Algorithm 3 can be used to find a suitable assignment $s(R)$ when $R \subseteq A$.

Let R_A, R_B, R_C, and R_D be a partition of the rows of A' corresponding to constraint sets (5.3), (5.4), (5.5), and upper bound from the standard linear programming relaxation ($x_{isd} \leq 1 \; \forall \; i = 1, \ldots, L, \; s \in S, \; d \in D_s$), respectively. If two rows of the matrix have a non-zero entry in the same column, we refer to them as *overlapping*. For a row r with nonzero entries in columns C, r is said to *overlap* the entries in columns C of another row. The matrix A has a column for each combination of (i, s, d) for $i = 1, \ldots, L$, $s \in S$, $d \in D_s$ corresponding to the variable x_{isd}. Let each (i, s, d) be a unique identifier for each column. Using this notation, two rows r_1 and r_2 are overlapping if $\beta_1 \cap \beta_2 \neq \emptyset$ where β_i represents the set of columns with nonzero entries in row r_i. Given subset of rows (either R_A, R_B, R_C, or R_D), each row corresponds to a constraint whose variables have at least one subscript in common and no other constraint in the subset has a variables with the same subscript. We say a row corresponds to the unique (to the subset of rows) subscript(s). The constraints of the MOMM model lead to some properties of A' that are used in the proof that A is TU:

1. Each row of A' has $0, 1$ entries.

2. Given any two rows in R_A, any two rows in R_B, or any two rows in R_C, the rows do not overlap.

3. Given a row r_C in R_C, r_C overlaps with at most one row in R_A. Moreover, for every column that r_C has an entry 1, there is an entry of 1 in the overlapping row in R_A.

4. For a row $r_C \in R_C$, a row $r_A \in R_A$, and two rows $r_B^1, r_B^2 \in R_B$ that overlap with r_C, if r_B^1 and r_A are overlapping, then r_B^2 and r_A are overlapping.

5. The rows of R_D make an identity matrix. Thus, no two rows in R_D overlap.

We describe why the properties of A' are true.

1. In each constraint, a variable appears at most once and if it appears, it has a coefficient of 1. Thus, the entries in A are 0,1, A' has 0,1 entries, and each row in A' has 0,1 entries.

2. A row in R_A corresponds to a unique $d \in D$, and the columns of that row with nonzero entries are $\{(i, s, d) \; \forall \; i = 1, \ldots, L, \; s \in S_d\}$. Thus, given a row corresponding to d^1 and another corresponding to d^2, $d^1 \neq d^2$ and we have $\{(i, s, d^1) \; \forall \; i = 1, \ldots, L, \; s \in S_{d^1}\} \cap \{(i, s, d^2) \; \forall \; i = 1, \ldots, L, \; s \in S_{d^2}\} = \emptyset$. A row in R_B corresponds to a unique pair of $i \in \{1, \ldots, L\}$ and $s \in S$, and the columns of that row with nonzero entries are $\{(i, s, d) \; \forall \; d \in D_s\}$. Thus, given a row corresponding to i^1, s^1 and another corresponding to i^2, s^2, $(i^1, s^1) \neq (i^2, s^2)$, we have $\{(i^1, s^1, d) \; \forall \; d \in D_{s^1}\} \cap \{(i^2, s^2, d) \; \forall \; d \in D_{s^2}\} = \emptyset$. A row in R_C corresponds to a unique pair of $d \in D$ and $s \in S_d$, and the columns of that row with nonzero entries are $\{(i, s, d) \; \forall \; i = 1, \ldots, L\}$. Thus, given a row corresponding to d^1, s^1 and another corresponding to d^2, s^2, $(d^1, s^1) \neq (d^2, s^2)$, we have $\{(i, s^1, d^1) \; \forall \; i = 1, \ldots, L\} \cap \{(i, s^2, d^2) \; \forall \; i = 1, \ldots, L\} = \emptyset$.

3. Each row in R_A has coefficient of one in columns $\{(i, s, d) \; \forall \; i = 1, \ldots, L, \; s \in S_d\}$ for a $d \in D$ unique to that row. Each row in R_C has an entry in columns $\{(i, s, d) \; \forall \; i = 1, \ldots, L\}$ for $d \in D$ and $s \in S_d$ unique to that row. For a given row of R_C corresponding to s and d, there is at most one row in R_A that corresponds to the same d. Moreover, $\{(i, s, d) \; \forall \; i = 1, \ldots, L\} \subseteq \{(i, s', d) \; \forall \; i = 1, \ldots, L, \; s' \in S_d\}$ for each $d \in D$ and $s \in S_d$. So, for every column with an entry of one in the a row from R_C, there is an entry of one in the overlapping row from R_A as we know $s \in S_d$.

4. The row r_C corresponding to a $d \in D$ and a $s \in S_d$ has nonzero entries in columns

$\{(i, s, d) \; \forall \; i = 1, \ldots, L\}$. Let r_B^1 and r_B^2 be two rows within R_B that overlap with r_C. Both $r_B^1,\ r_B^2$ must correspond to s, otherwise the rows do not overlap with r_C. Then r_B^1 and r_B^2 differ due to index $i^1 \in \{1, \ldots, L\}$ and $i^2 \in \{1, \ldots, L\}$ to which they correspond, respectively. The columns of r_B^1 with a nonzero entry are $\{(i^1, s, d') \; \forall \; d' \in D_s\}$. The columns of r_B^2 with a nonzero entry are $\{(i^2, s, d') \; \forall \; d' \in D_s\}$. Each row $r_A \in R_A$ is defined for a specific $d^* \in D$ and has nonzero elements in columns $\{(i, s, d^*) \; \forall \; i = 1, \ldots, L,\ s \in S_{d^*}\}$. If $\{(i^1, s, d') \; \forall \; d' \in D_s\} \cap \{(i, s, d^*) \; \forall \; i = 1, \ldots, L,\ s \in S_{d^*}\} \neq \emptyset$ then $\{(i^2, s, d') \; \forall \; d' \in D_s\} \cap \{(i, s, d^*) \; \forall \; i = 1, \ldots, L,\ s \in S_{d^*}\} \neq \emptyset$ as $i^1,\ i^2 \in \{1, \ldots, L\}$.

5. Each row of R_D corresponds to a unique (i, s, d) for $i \in \{1, \ldots, L\}$, $s \in S$, $d \in D_s$ and has a single nonzero entry in column (i, s, d).

Given these facts, we construct Algorithm 3, which assigns $s(R)$ values in such a way that satisfies the Ghouila-Houri Theorem. Algorithm 3 begins with a row vector of zeros, **t**, with length equal to the number of columns in A'. We iterate through the rows in R sorted in the order of $R_A,\ R_B,\ R_C,\ R_D$. For each row $r \in R$, if adding the row to **t** would result in a 2 entry in the row vector, then we set $s(r) = -1$, otherwise set $s(r) = 1$. Then add $s(r)r$ to **t**.

Algorithm 3 $s(R) : R \to \pm 1,\ R \subseteq A'$

t $\leftarrow$ **0** ▷ A row vector to store $\sum_{i=1,\ldots,k-1} s(r_i)r_i$
$R \leftarrow R$ sorted so rows appear in order R_A, R_B, R_C, R_D
for **r** $\in R$ **do** ▷ For each row vector **r** in R
    if 2 an entry in $\mathbf{t}^T + \mathbf{r}$ **then**
        $s(\mathbf{r}) = -1$
    else
        $s(\mathbf{r}) = 1$
    end if
    $\mathbf{t} = \mathbf{t} + s(\mathbf{r})\mathbf{r}$
end for

Through a series of observations, we show how to create an assignment $s(R)$ that satisfies the Ghouila-Houri Theorem for any row $R \subseteq A$. Observation 1 states that if **t** had

0,1 entries to begin an iteration of Algorithm 3, then **t** had 0,1 entries after the iteration if a single condition is met. In Observation 2, we show this condition is met during each iteration of Algorithm 3. Observation 3 states that Algorithm 3 creates an assignment $s(R')$ that satisfies the Ghouila-Houri Theorem for any row $R' \subseteq A'$. Observation 4 states that we can construct an assignment $s(R)$ that satisfies the Ghouila-Houri Theorem for any row $R \subseteq A$. We begin with Observation 1.

Observation 1: Given $R \subseteq A'$, let r_k be the k^{th} row considered by Algorithm 3. Let C_k be the columns of r_k which have nonzero entries. Let $\mathbf{t} = \sum_{i=1\ldots k-1} s(r_i)r_i$ have entries in $\{-1, 0, 1\}$. Let $s(r_k)$ be the assigned value given by Algorithm 3. If for r_k there does not exist two columns $c^1,\ c^2 \in C_k$ with entries of -1 and 1 in **t**, then $\mathbf{t} + s(r_k)r_k$ has entries in $\{-1, 0, 1\}$.

Proof: The entries in columns not in C_k are the same in **t** and $\mathbf{t} + s(r_k)r_k$ for any $s(r_k)$, the entries in columns not in C_k of r_k are zero by definition of C_k. Thus, we focus on the entries in columns C_k. Note $\mathbf{t} + r_k$ is equivalent to adding one to the entries in columns C_k of **t** as r_k is $0, 1$ valued by property 1. Similarly, $\mathbf{t} - r_k$ is equivalent to subtracting one to the entries in columns C_k of **t**.

If $s(r_k) = 1$, then the highest entry in any columns in C_k of **t** is 0. If the highest entry was 1, then 2 is an entry of $\mathbf{t} + r_k$ and Algorithm 3 would have set $s(r_k) = -1$. Thus, adding one to the entry in any column in C_k of **t** does not result in an entry in any column more than 1. No entry in $\mathbf{t} + s(r_k)r_k$ is lower than -1 as the lowest value in **t** is no less than -1.

If $s(r_k) = -1$, then the highest entry in columns C_k of **t** is 1. If the highest entry was not 1 (either -1 or 0), 2 would not be an entry of $\mathbf{t} + r_k$ and Algorithm 3 would set $s(r_k) = 1$. If no two columns $c^1,\ c^2$ of **t** have entries of -1 and 1, the lowest value in columns C_k of **t** is 0 and subtracting 1 from the entry in columns C_k of **t** results in entries no less than -1. No entry in $\mathbf{t} + s(r_k)r_k$ is higher than 1 as the highest value in **t** is no greater than 1.

□

Observation 2 shows this condition is met during each iteration of Algorithm 3.

Observation 2: Given $R \subseteq A'$, let r_k be the k^{th} row considered by Algorithm 3. Let C_k be the columns of r_k that have nonzero entries. Let $\mathbf{t} = \sum_{i=1\ldots k-1} s(r_i)r_i$ have entries in $\{-1, 0, 1\}$. For r_k there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2.

Proof: If $r_k \in R_A$, then the entries in columns C_k of $\mathbf{t}$ are all zeros as no two rows of R_A overlap, the rows of R_A are visited first, and $\mathbf{t}$ begins as a zero vector. Thus, there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2. Note that if $r_k \in R_A$, $s(r_k) = 1$ as r_k overlaps with no row in $\{r_i,\ i = 1, \ldots, k-1\}$ by property 2. Thus, in columns C_k, $\sum_{i=1\ldots k-1} s(r_i)r_i$ has entries of zeros and $r_k + \sum_{i=1\ldots k-1} s(r_i)r_i$ does not have an entry of 2 as $\sum_{i=1\ldots k-1} s(r_i)r_i$ did not have an entry of 2.

If $r_k \in R_B$, then the entries in columns C_k of $\mathbf{t}$ must be greater or equal to zero. This relies on the fact that $s(r) = 1$ for each $r \in R \cap R_A$, and no two rows in R_B overlap, by property 2. Thus, there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2.

If $r_k \in R_C$, r_k overlaps with at most one row r_A from R_A. There are two cases $r_A \in R$ or $r_A \notin R$. We consider the first: $r_A \in R$. We have shown $s(r_A) = 1$. By property 1 and 3, the entry in columns C_k in r_A is one. Since no two rows in R_B overlap, rows of R_B are $0, 1$ valued by property 1, and no two rows in R_C overlap, the entry in every column in C_k of $\mathbf{t}$ is at least zero. Thus, there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2. We consider the second: $r_A \notin R$. By property 3, r_k overlaps with no other row in R_A. As a result, each entry in columns C_k in $\sum_{r \in R \cap R_A} s(r)r$ must be zero. Rows of $R \cap R_B$ that are not overlapping with r_k do not change the values in columns C_k of $\sum_{i=1\ldots k-1} s(r_i)r_i$ by the definition of overlapping. If no row from R_B is in R, then the entries in columns C_k of $\mathbf{t}$ must be zero. If a single row in from R_B is in R, then all nonzero entries in $\mathbf{t}$ will be of the same sign as all nonzero entries in a row are a value of one by property

one, and two columns with entries of -1 and 1 do not exist. Suppose more than one row from R_B is in R. Consider any two rows $r_B^1, r_B^2 \in R \cap R_B$ which overlap with r_k. Let r_B^1 be the m^{th} row considered in Algorithm 3. We know $r_B^1 + \sum_{i=1,\dots,m-1} s(r_i)r_i$ has an entry of 2 if and only if $r_B^1 + \sum_{r \in R \cap R_A} s(r)r$ has an entry of 2, as no two rows in R_B overlap by property 2. Moreover, $r_B^1 + \sum_{r \in R \cap R_A} s(r)r$ has an entry of 2 if and only if a row $\hat{r}_A \in R_A$ overlapping row r_B^1 is in R. Thus, by property 4, r_B^2 overlaps with $\hat{r}_A$ and $r_B^2 + \sum_{r \in R \cap R_A} s(r)r$ has an entry of 2 As a result, each $r_B \in R \cap R_B$ overlapping r_k is assigned the same $s(r_B)$ value and the nonzero entries in columns C_k of $\mathbf{t}$ are equal. Thus, there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2.

If $r_k \in R_D$, then r_k only has only a single column with a nonzero entry by property 5. Thus, $|C_k| = 1$, and there does not exist two columns $c^1,\ c^2$ in C_k where $\mathbf{t}$ has entries of -1 and 1 in c^1 and c^2.

□

Given Observation 1 and 2 are true, Observation 3 states that Algorithm 3 creates an assignment $s(R)$ that satisfies the Ghouila-Houri Theorem for a $R \subseteq A'$.

Observation 3: Given $R \subseteq A'$, Algorithm 3 finds an assignment $s(R) : R \to \pm 1 : \sum_{r \in R} s(r)r$ with entries in entries in $\{-1, 0, 1\}$.

Proof: Let $\mathbf{t}_k$ be the value of $\mathbf{t}$ in Algorithm 3 when the k^{th} row is considered. The row vector $\mathbf{t}_1$ is initialized with entries in $\{0\} \subset \{-1, 0, 1\}$. For the k^{th} visited row $r_k \in R$, $\mathbf{t}_{k+1} = \mathbf{t}_k + s(r_k)r_k$ has entries in $\{-1, 0, 1\}$ by Statements 1 and 2. Thus, after the final row in R is considered $\mathbf{t}_{|R|+1}$ has entries in $\{-1, 0, 1\}$ and $\mathbf{t}_{|R|+1} = \mathbf{t}_{|R|} + s(r_{|R|})r_{|R|} = 0 + \sum_{i=1,\dots,|R|} s(r_i)r_i = \sum_{r \in R} s(r)r$.

□

As noted earlier, Observation 3 only applies to cases in which $R \subseteq A' \subset A$. In Observation 4, we extend this to the case in which $R \subseteq A$. The only difference between A and A' is the inclusion of rows corresponding to constraint set (5.2), denoted R_E. The matrix A is

defined such that all constraints are written in the form $Ax \leq b$. As a result, the rows in R_E are $0, -1$ valued, and for each row $r_E \in R_E$, there exists a row $r_A \in R_A$ such that $r_E = -r_A$, and vice versa. Furthermore, a row $r_E \in R_E$ overlaps with exactly one row in R_A and vice versa as the rows in R_A do not overlap.

Given $R \subseteq A$, if $R \subseteq A'$, then we can find a valid $s(R)$ using Algorithm 3, and by Statement 3, this assignment satisfies the condition that $\sum_{r \in R} s(r)r$ has entries in $\{-1, 0, 1\}$. Thus, we consider the case that R includes at least one row from R_E.

For every row $r_E \in R_E$ included in R, the row $r_A \in R_A$ such that $r_A = -r_E$ is either in R or not. Let R_E^{In}, R_A^{In} denote the set of rows of R_E and R_A, respectively, such that both $r_E \in R$ and $r_A \in R$. Let R_E^{Out}, R_A^{Out} denote the set of rows of R_E and R_A, respectively, such that $r_E \in R$ but $r_A \notin R$. Note that $R_E = R_E^{\text{In}} \cup R_E^{\text{Out}}$ as either the row $r_A = -r_E$ is in R or not. Create a set $R' = (R \cup R_A^{\text{Out}}) \setminus (R_E \cup R_A^{\text{In}}) \subseteq A'$. Run Algorithm 3 on R' to obtain $\hat{s}(R')$ such that the entries of $\sum_{r \in R} \hat{s}(r)r$ are in $\{-1, 0, 1\}$ by Statement 3.

For rows $r \in R \setminus R_E$, let $s(r) = \hat{s}(r)$. For each row $r_E \in R_E^{\text{In}}$ and the overlapping row $r_A \in R_A^{\text{In}}$, let $s(r_E) = s(r_A) = 1$. Note that $s(r_E)r_E + s(r_A)r_A = r_E + r_A = r_E - r_E = 0$. For each row $r_E \in R_E^{\text{Out}}$ overlapping row $r_A \in R_A^{\text{Out}}$, let $s(r_E) = -\hat{s}(r_A)$. Note that $s(r_E)r_E = -\hat{s}(r_A)r_E = \hat{s}(r_A)r_A$. This defines an assignment $s(R)$.

Observation 4: Given the assignment $s(R)$ just described, $\sum_{r \in R} s(r)r$ has entries in $\{-1, 0, 1\}$ for any $R \subseteq A$.

Proof:

$$\begin{aligned}
\sum_{r \in R} s(r)r &= \sum_{r \in R_E^{\text{In}}} s(r)r + \sum_{r \in R_A^{\text{In}}} s(r)r + \sum_{r \in R_E^{\text{Out}}} s(r)r + \sum_{r \in R \setminus (R_A^{\text{In}} \cup R_E)} s(r)r \\
&= \sum_{r \in R_E^{\text{In}}} s(r)r + \sum_{r \in R_E^{\text{In}}} -s(r)r + \sum_{r \in R_E^{\text{Out}}} s(r)r + \sum_{r \in R \setminus (R_A^{\text{In}} \cup R_E)} s(r)r \\
&= \sum_{r \in R_E^{\text{Out}}} s(r)r + \sum_{r \in R \setminus (R_A^{\text{In}} \cup R_E)} s(r)r
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{r \in R_A^{\text{Out}}} \hat{s}(r) r + \sum_{r \in R \setminus (R_A^{\text{In}} \cup R_E)} \hat{s}(r) r \\
&= \sum_{r \in (R \cup R_A^{\text{Out}}) \setminus (R_A^{\text{In}} \cup R_E)} \hat{s}(r) r \\
&= \sum_{r \in R'} \hat{s}(r) r
\end{aligned}
$$

□ □

Proof to Theorem 5.2. We first describe how to construct a network flow problem and then show the minimum cost network flow (MCNF) provides an equivalent solution as the MOMM problem. Let the MCNF network be G with nodes N and arcs A. We let $N = \{a, b\} \cup N_1 \cup N_2 \cup N_3$, where a is a source node and b is a sink node, and $A = \{(b, a)\} \cup A_1 \cup A_2 \cup A_3 \cup A_4$. We let (b, a) denote a directed arc with a tail at b and a head at a. The flow on the arc (b, a) must be no less than zero. Each A_i for $i = 1, 2, 3, 4$ represent a subset of the arcs where each A_i is mutually exclusive, and N_i for $i = 1, 2, 3$ represent a subset the nodes where each N_i is mutually exclusive. We describe how to build these subsets.

For each element in $d \in D$ create a node labeled d and add to N_1. For each $d \in N_1$, create an arc (a, d) and add to A_1. The flow on each of these arcs must be at least J_d^ℓ, but no more than J_d^u. This results in $|A_1| = |N_1| = |D|$. For each node $d \in N_1$ and for each element in $s \in S_d$ create a node labeled s_d and add to N_2, and create an arc (d, s_d) and add to A_2. The flow on each arc in A_2 must be at least zero but no more than one. This is $|A_2| = |N_2| = \sum_{d \in D} |S_d| \leq |D||S|$ nodes and arcs. For each element $s \in S$ and each $i = 1, \ldots, L$, create a node labeled i_s and add to N_3. For each $d \in D$, $s \in S_d$, $i = 1, \ldots, L$, create an arc (s_d, i_s) and add to A_3. The flow on each arc in A_3 must be at least zero but no more than one. This results in $|N_3| = |S|L$ and $|A_3| = \sum_{d \in D} |S_d|L \leq |D||S|L$. For each node $i_s \in N_3$, create an arc (i_s, b) and add to A_4. The flow on each arc in A_4 must be at least zero but no more than one. This is $|A_4| = |S||$ arcs. The value of the flow is zero for every

arc except those in A_3. The value of the flow on an arc $(s_d, i_s) \in A_3$ is $-w_i y_d u_{sd}$ for each $i = 1, \ldots, L$, $d \in D$, $s \in S_d$.

For example, if $D = \{A, B\}$, $S_A = \{X, Y\}$, $S_B = \{Y, Z\}$, and $L = 2$ then:

$$N_1 = \{A, B\}$$

$$A_1 = \{(a, A), (a, B)\}$$

$$N_2 = \{X_A, Y_A, Y_B, Z_B\}$$

$$A_2 = \{(A, X_A), (A, Y_A), (B, Y_B), (B, Z_B)\}$$

$$N_3 = \{1_X, 2_X, 1_Y, 2_Y, 1_Z, 2_Z\}$$

$$A_2 = \{(X_A, 1_X), (X_A, 2_X), (Y_A, 1_Y), (Y_A, 2_Y), (Y_B, 1_Y), (Y_B, 2_Y), (Z_B, 1_Z), (Z_B, 2_Z)\}$$

$$A_4 = \{(1_X, b), (2_X, b), (1_Y, b), (2_Y, b), (1_Z, b), (2_Z, b)\}$$

See Figure 7.2 for a visualization of this network.

In total this network has $|D| + \sum_{d \in D} |S_d| + \sum_{d \in D} |S_d| L + |S| L + 1 \leq 4|D||S|L + 1$ arcs and $|D| + \sum_{d \in D} |S_d| + |S| L + 2 \leq (|S| + 1)|D| + |S| L + 2$ nodes which is polynomial in the number of variables in the MOMM problem, $\mathcal{O}(|D||S|L)$.

To construct the network flow problem, we introduce a flow variable $y_{\alpha\beta}$ for every arc between nodes α and β in the network. The formulation of the MNCF problem is then:

$$\min_{y} \sum_{i=1}^{L} \sum_{s \in S} \sum_{d \in D_s} -w_i v_d u_{sd} y_{s_d i_s} \tag{7.10}$$

$$s.t. \sum_{\beta \in N : (\beta, \alpha) \in A} y_{\beta\alpha} = \sum_{\beta \in N : (\alpha, \beta) \in A} y_{\alpha\beta} \quad \forall\, \alpha \in N \tag{7.11}$$

$$y_{ad} \geq J_d^{\ell} \quad \forall\, d \in D \tag{7.12}$$

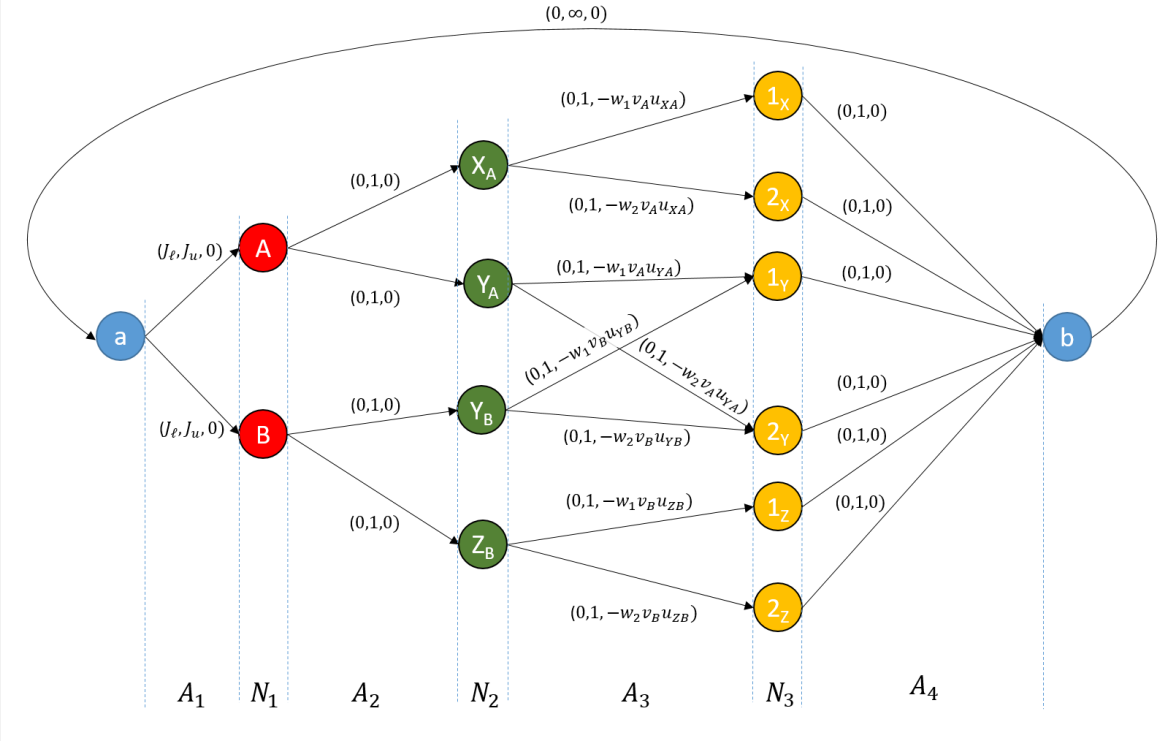


Figure 7.2: Network diagram for $D = \{A, B\}$, $S_A = \{X, Y\}$, $S_B = \{Y, Z\}$, and $L = 2$. Tuples on each arc describe the flow (lower bound, upper bound, objective coefficient), respectively.

$$y_{ad} \leq J_d^u \quad \forall\, d \in D \tag{7.13}$$

$$y_{ds_d} \leq 1 \quad \forall\, d \in D,\ s \in S_d \tag{7.14}$$

$$y_{s_d i_s} \leq 1 \quad \forall\, d \in D,\ s \in S_d,\ i = 1, \ldots, L \tag{7.15}$$

$$y_{i_s b} \leq 1 \quad \forall\, s \in S,\ i = 1, \ldots, L \tag{7.16}$$

$$y_{\alpha\beta} \geq 0 \quad \forall\, (\alpha, \beta) \in A \tag{7.17}$$

The objective (7.10) is to minimize the value of the flow over the arcs (s_d, i_s) for $d \in D$, $s \in S_d$, and $i = 1, \ldots, L$, where the value of each unit of flow is $-w_i v_d u_{sd}$. The flow entering and leaving each node must be balanced, constraint set (7.11). The flow on each arc in section 1 has a lower bound of J_d^ℓ and an upper bound of J_d^u, constraint sets (7.12) and

(7.13). The flow on each arc in section 2 has an upper bound of one, constraint set (7.14). The flow on each arc in section 3 has an upper bound of one, constraint set (7.15). The flow on each arc in section 4 has an upper bound of 1, (7.16). The flow on arc (b, a) has no upper bound. The lower bound on the flow of every arc is zero, constraint set (7.17). This has a polynomial number of variables and constraints with respect to the number of variables in the MOMM problem, $\mathcal{O}(|D||S|L)$ as there are $\mathcal{O}(|D||S|L)$ arcs and $\mathcal{O}(|S||D|)$ nodes in the MCNF problem.

We show that the MOMM problem can be reduced to a MCNF problem. To do so, we show that given an optimal solution to the MCNF instance, a solution of equivalent value can be constructed to the MOMM instance. Likewise, given a feasible solution to the MOMM problem, we can construct a feasible solution to the MCNF problem with equivalent value. Be begin with the latter.

Assume we have a feasible solution to the MOMM problem, x_{isd} for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$. We construct a feasible solution to the MCNF problem starting with the flows associated with section 3 arcs. Let $y_{s_d i_s} = x_{isd}$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$. The flow balance constraints determine the remaining flow values. Thus the remaining flow values are:

$$y_{ad} = \sum_{s \in S_d} y_{d,s_d} = \sum_{s \in S_d} \sum_{i=1}^{L} y_{s_d,i_s} = \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} \quad \forall\, d \in D$$

$$y_{ds_d} = \sum_{i=1}^{L} y_{s_d,i_s} = \sum_{i=1}^{L} x_{isd} \quad \forall\, d \in D,\ s \in S_d$$

$$y_{i_s,b} = \sum_{d \in D_s} y_{s_d i_s} = \sum_{d \in D_s} x_{isd} \quad \forall\, i = 1, \ldots, L,\ s \in S$$

$$y_{ba} = \sum_{s \in S} \sum_{i=1}^{L} y_{i_s,b} = \sum_{s \in S} \sum_{i=1}^{L} \sum_{d \in D_s} y_{s_d i_s} = \sum_{d \in D} \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd}$$

We show all constraints in the MCNF problem are satisfied. We begin with the flow balance constraints (7.11). Specifically, we show that for every node, the inflow equals the

outflow.

We begin with node a:

$$\textbf{Inflow:} \quad y_{ba} = \sum_{d \in D} \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd}$$

$$\textbf{Outflow:} \quad \sum_{d \in D} y_{ad} = \sum_{d \in D} \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd}$$

Next the nodes in N_1. For a given $d \in D$, the inflow and outflow at node $d \in N$:

$$\textbf{Inflow:} \quad y_{ad} = \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd}$$

$$\textbf{Outflow:} \quad \sum_{s \in S_d} y_{ds_d} = \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd}$$

Next the nodes in N_2. For a given $d \in D$ and $s \in S$, the inflow and outflow at node $s_d \in N$:

$$\textbf{Inflow:} \quad y_{ds_d} = \sum_{i=1}^{L} x_{isd}$$

$$\textbf{Outflow:} \quad \sum_{i=1}^{L} y_{s_d i_s} = \sum_{i=1}^{L} x_{isd}$$

Next the nodes in N_3. For a given $s \in S$ and $i = 1, \ldots, L$, the inflow and outflow at node $i_s \in N$:

$$\textbf{Inflow:} \quad \sum_{d \in D_s} y_{s_d i_s} = \sum_{d \in D_s} x_{isd}$$

$$\textbf{Outflow:} \quad y_{i_s,b} = \sum_{d \in D_s} x_{isd}$$

And finally, node b:

$$\textbf{Inflow:} \quad \sum_{s \in S} \sum_{i=1}^{L} y_{i_s,b} = \sum_{s \in S} \sum_{i=1}^{L} \sum_{d \in D_s} x_{isd}$$

$$\textbf{Outflow:} \quad y_{ba} = \sum_{s \in S} \sum_{i=1}^{L} \sum_{d \in D_s} x_{isd}$$

We next verify that constraints (7.12) - (7.17) are satisfied.

Constraint Set (7.12): $y_{ad} = \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} \geq J_d^{\ell}$ for all $d \in D$ where the final inequality holds due to constraint set (5.2) which defines a feasible solution to the MOMM problem.

Constraint Set (7.13): $y_{ad} = \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} \leq J_d^{u}$ for all $d \in D$ where the final inequality holds due to constraint set (5.3) which defines a feasible solution to the MOMM problem.

Constraint Set (7.14): $y_{ds_d} = \sum_{i=1}^{L} x_{isd} \leq 1$ for all $d \in D,\ s \in S_d$ where the final inequality holds due to constraint set (5.5) which defines a feasible solution to the MOMM problem.

Constraint Set (7.15): $y_{s_d i_s} = x_{isd} \leq 1$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$ where the final inequality holds due to constraint set (5.6) which defines a feasible solution to the MOMM problem.

Constraint Set (7.16) $y_{i_s,b} = \sum_{d \in D_s} x_{isd} \leq 1$ for all $i = 1, \ldots, L,\ s \in S$ where the final inequality holds due to constraint set (5.4) which defines a feasible solution to the MOMM problem.

Constraint Set (7.17): Each the value of each flow variable is determined by a sum of non-negative x_{isd} variables with nonnegative coefficients. Thus, the flow cannot be less than zero.

Thus, given a solution to the MOMM problem, we can construct a feasible solution to an instance of the MCNF problem. Finally, we show the feasible solution provides an equivalent (opposite sign) objective function value.

$$\begin{aligned} \sum_{i=1}^{L} \sum_{s \in S} \sum_{d \in D_s} -w_i v_d u_{sd} y_{s_d i_s} &= -[\sum_{i=1}^{L} \sum_{s \in S} \sum_{d \in D_s} w_i v_d u_{sd} y_{s_d i_s}] \\ &= -[\sum_{i=1}^{L} \sum_{s \in S} \sum_{d \in D_s} w_i v_d u_{sd} x_{isd}] \end{aligned}$$

Next, we must show that given an optimal solution to the MCNF problem, a feasible solution to the MOMM problem of equivalent value can be constructed. When $J_d^\ell \in \mathbb{Z}$ and $J_d^u \in \mathbb{Z}$, then there exists an optimal solution to the MCNF where the flows take integer values as the constraint matrix of the MCNF problem is TU. Let the value of the flows at this optimal solution be integer. Let $x_{isd} = y_{s_d i_s}$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$. We show that every constraint in the MOMM model is satisfied and the value of the MOMM solution is equivalent to the solution of the MCNF solution.

Constraint set (5.2) and (5.3) in the MOMM formation requires that $J_d^\ell \leq \sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} \leq J_d^u$ for each $d \in D$. By flow balance constraints:

$$\sum_{s \in S_d} \sum_{i=1}^{L} x_{isd} = \sum_{s \in S_d} \sum_{i=1}^{L} y_{s_d, i_s} = \sum_{s \in S_d} y_{d, s_d} = y_{ad}$$

Since we begin at a feasible solution for the MCNF instance, constraint sets (7.12) and (7.13) ensure, $J_d^\ell \leq y_{ad} \leq J_d^u$.

Constraint set (5.4) requires $\sum_{d \in D_s} x_{isd} \leq 1$ for each $i = 1, \ldots, L,\ s \in S$. By flow balance constraints:

$$\sum_{d \in D_s} x_{isd} = \sum_{d \in D_s} y_{s_d i_s} = y_{i_s, b}$$

Constraint set (7.16) ensures $y_{i_s, b} \leq 1$.

Constraint set (5.5) requires $\sum_{i=1}^{L} x_{isd} \leq 1$ for each $d \in D,\ s \in S_d$. By flow balance constraints:

$$\sum_{i=1}^{L} x_{isd} = \sum_{i=1}^{L} y_{s_d, i_s} = y_{d s_d} \quad \forall\, d \in D,\ s \in S_d$$

Constraint set (7.14) ensures $y_{d s_d} \leq 1$.

By constraint sets (7.15) and (7.17) in the MCNF formulation and the fact that $y_{s_d i_s}$ for all $d \in D,\ s \in S_d$ is integer, and thus $x_{isd} \in \{0, 1\}$.

Finally, we show the feasible solution provides an equivalent (opposite sign) objective

function value.

$$\begin{aligned}\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} x_{isd} &= -[\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} -w_i v_d u_{sd} x_{isd}] \\ &= -[\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} -w_i v_d u_{sd} y_{s_d i_s}]\end{aligned}$$

Thus for an optimal solution to the MCNF problem, we can construct a feasible solution to the MOMM problem of equivalent value using the relation $x_{isd} = y_{s_d i_s}$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$.

Thus, the MOMM can be reduced to a MCNF formulation. Given an optimal solution y^* to the MCNF formulation, we can construct a solution x that is optimal to the MOMM problem where $x_{isd} = y_{s_d i_s}$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$. Too see why this is true, suppose x is not optimal to the MOMM problem. Since x is not optimal to the MOMM instance, there exists a x^* feasible to the MOMM problem such that $\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} x^*_{isd} > \sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} x_{isd}$. Then, we have shown we can construct a solution $\hat{y}$, where $\hat{y}_{s_d i_s} = x^*_{isd}$ for all $i = 1, \ldots, L$, $s \in S$, $d \in D_s$, such that:

$$\begin{aligned}\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} -w_i v_d u_{sd} \hat{y}_{s_d i_s} &= -[\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} \hat{y}_{s_d i_s}] \\ &= -[\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} x^*_{isd}] \\ &< -[\sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} w_i v_d u_{sd} x_{isd}] \\ &= \sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} -w_i v_d u_{sd} x_{isd} \\ &= \sum_{i=1}^{L}\sum_{s\in S}\sum_{d\in D_s} -w_i v_d u_{sd} y^*_{s_d i_s}\end{aligned}$$

which is a contradiction on our assumption that y^* was optimal.

Thus the MOMM problem can be reduced to a MCNF problem in polynomial time as

MCNF instance has $\mathcal{O}(|D||S|L)$ nodes and arcs, the MCNF problem can be solved using linear programming, and thus the MOMM problem can be solved in polynomial time with respects to the inputs D, S, and L. □

□

Appendix 8

Chapter 3 Supplemental Material

8.1 Generalized contiguity constraints

In some cases, the definition of contiguity used within the PLCP may be overly restrictive, and election officials may be willing to accept a more generalized definition of contiguity. For example, in Richland County, the law states that if the Board of Voter Registration and Elections of Richland County determines that a precinct contains no suitable location for a polling place, the board may locate the polling place inside the county and within five miles of the precinct's boundaries. A strict interpretation of this constraint can be enforced within the PLCP by replacing constraint set (3.11) with the constraints

$$x_{ij} = 0 \quad \forall\, i \in I, j \in J \setminus J_i \tag{8.1}$$

where J_i are the polling locations further than the allowable distance from the population district (e.g., 5 miles from the border of a population district) such that the population district cannot be assigned to the polling locations in J_i. Note that the resulting district do not need to be contiguous.

However, these constraints can be very restrictive in cases where consolidation is necessary. We present a modification to constraint set (3.11) that requires at least one population district of a contiguous polling location subdistrict to be within a distance D from the polling location. Let $D_j \subseteq I$ be the subset of population districts with a portion of their boundaries within the required distance from a polling location with $\{i(j)\} \subseteq D_j$. We can then replace constraint set (3.11) with the following constraints:

$$x_{ij} \leq \sum_{k \in D_j \cap N_i} x_{kj} \qquad \forall\, j \in J, i \in I \setminus D_j : N_i \cap D_j \neq \emptyset \tag{8.2}$$

$$x_{ij} \leq \sum_{k \in \hat{C}} x_{kj} \qquad \forall\, j \in J, i \in I \setminus D_j, \hat{C} \tag{8.3}$$

where $\hat{C}$ represents an (i, j)-separator according to the updated definition of contiguity. Constraint set (8.3) can be enforced using lazy constraints similar to those discussed within the main text by adjusting Algorithm 1. To update this procedure, we simply modify line 4 in Algorithm 1 to $T_j := \{i \in I : x^*_{ij} = 1\} \cup \{D_j\}$.

8.2 Special-case contiguity guarantee

Theorem 8.1 provides conditions under which we can guarantee that the contiguity constraints are redundant within the PLCP and an optimal solution can be identified without enforcing them. However, finding a ranking that satisfies the conditions of Theorem 8.1 is non-trivial due to the non-uniqueness of the rank of polling locations.

Theorem 8.1. *Suppose that there is an instance for the PLCP where constraint sets* (3.7)-(3.9) *define redundant constraints. Let $q_i(j)$ be the rank of each polling location $j \in J$ for each population district $i \in I$. For each $i \in I$, let $J(i) = J \setminus (\hat{J} \cup \{j(i)\} \cup \{j \in J : i(j) \in N_i\})$. If for each $i \in I$ and $j \in J(i)$ there exists a path P_{ij} in the contiguity graph G from i to $i(j)$ such that for each $i' \in P_{ij} \setminus \{i, i(j)\}$ and for all $j' \in \{j \in J \setminus \hat{J} : q_i(j') > q_i(j)\}$ it holds that $q_{i'}(j') > q_{i'}(j)$, then*

there exists an optimal solution $x^* \in \{0,1\}^{|I|\times|J|}$ *that implicitly satisfies the constraint set* (3.11).

Proof to Theorem 8.1. Suppose there does not exist such an optimal solution x^*. Then for all optimal solutions, $x \in \{0,1\}^{|I|\times|J|}$, there exists an $i \in I$ assigned to $j \in N \setminus \hat{J}$ such that $i \notin \{i(j)\} \cup N_{i(j)}$ between which there exists an (i,j)-separator, C, such that:

$$x_{ij} > \sum_{k \in C} x_{kj}$$

Let i' be a node in $P_{ij} \setminus \{i, i(j)\}$ such that $x_{i'j'} = 1$ and $j' \neq j$. There must exist a i', otherwise the existence of P_{ij} contradicts the existence of an (i,j)-separator such that constraint set (3.11) is violated.

By assumption either $q_i(j') < q_i(j)$ or $q_{i'}(j) < q_{i'}(j')$. If the former case is true, then we can reassign i to j' without increasing the OFV by Definition 3.3. If the latter is true, then we can reassign i' to j without increasing the OFV by Definition 3.3. If in the resulting solution, for all $i \in I$ assigned to $j \in J \setminus \hat{J}$ such that $i \notin \{i(j)\} \cup N_{i(j)}$ there does not exist an (i,j)-separator, the solution is contiguous. If there does exist one, then we repeat the process. In each iteration the rank of the assigned polling location of each population district is strictly non-decreasing and there is at least one population district for which the rank of the assigned polling location strictly decreases. Thus, the solutions do not repeat. The resulting solution must be contiguous with an OFV no more than x^*. Thus, either x is not optimal or we have found a contiguous assignment that is also optimal. Both are contradictions to the initial assumptions. □

Appendix 9

Chapter 4 Supplemental Material

9.1 Heuristic Method Psuedocode

The pseudocode of the DBLP heuristic solution method is presented in Algorithm 4. We assume that the reformulation of the DBLP outlined in Section 4.4.1 is used throughout the heuristic. We also assume that q takes a value of either one or two; it can easily be extended to cases where q is larger. When $q = 0$, lines 2-7 can be replaced so that $x' \in \{0,1\}^{|E|}$ and $y' \in \{0,1\}^{|N|}$ are defined from a (heuristic) solution to the TSP over the set of locations T. The steps of the heuristic are as follows. The initial solution of drop box locations and the associated collection tour is found in lines 1-7. In lines 1-2, a CTP instance is solved based on the DBLP instance. In lines 3-7, a second CTP instance is created and solved. Solutions to both instances must be feasible for the respective CTP instances, but need not be optimal. The initial solution for the DBLP is represented by N^0, which represents the selected drop box locations, and $\mathcal{C}^0$, which represents the collection tour over the selected locations (lines 8-9).

In lines 11-38, Algorithm 4 finds DBLP solutions meeting a progressively higher bound r for the minimal access function value. In line 11, a set C is initialized to store previously

found solutions. This set is used later (lines 31-32) determine if Algorithm 4 has re-found a solution. In lines 15-18, Algorithm 4 determines which pairs of drop box locations are feasible by checking constraint sets (4.2) and (4.3). We do not allow $i = j$, since this would represent no change to the drop box system. In line 19, we estimate the change in the collection tour cost resulting from the removal of i and insertion of j, $\Delta\hat{c}(i,j)$. This can be estimated using a variety of methods, with the simplest being cheapest cost insertion and shortcut removal. In line 20, we calculate the change in the minimal access function value for pair (i,j), $\Delta r(i,j)$. In line 21, we identify the best feasible pair using an angle-based approach similar to that used in [48]. The pairs are assessed based on the angle, using a counter-clockwise orientation, between the vector $\langle -1, 0\rangle$ and the vector $\langle \Delta r(i,j), \Delta\hat{c}(i,j)\rangle$. The angle, $\theta_{i,j}$ can be calculated using the following formula:

$$\theta(\Delta r, \Delta\hat{c}) = \begin{cases} 2\pi - \cos^{-1}\left(\frac{-\Delta r}{\sqrt{\Delta r^2 + \Delta\hat{c}^2}}\right) & \Delta\hat{c} \geq 0 \\ \cos^{-1}\left(\frac{-\Delta r}{\sqrt{\Delta r^2 + \Delta\hat{c}^2}}\right) & \text{otherwise} \end{cases} \tag{9.1}$$

Algorithm 4 selects the pair with the lowest $\theta(\Delta r, \Delta\hat{c})$ in line 22; this leads to DBLP solutions with a lower cost. In lines 29-30, the incumbent solution is updated based on the selected pair to swap. In line 31-35, the minimal access function value, r, is updated for the next iteration. If Algorithm 4 has re-found a solution (lines 31-32), r is set to be the current minimum access function value. This avoids future 'cycling' where Algorithm 4 finds the same solution multiple times. If Algorithm 4 has found a new solution, then r is updated to be the minimum of $A_w(N^k)$ and $r + \varepsilon$. This ensures that the Algorithm 4 is able to find a feasible solution to the DBLP in the next iteration if one exists (this is a result of $v_{wn} > 0$ for all $n \in N \setminus T$ and $w \in W$). However, if ε is sufficiently small, r could be updated by setting $r = r + \varepsilon$. The value of ε is sufficiently small when there is guaranteed to be at least one feasible pair to swap in each iteration. The following ε value is guaranteed to be sufficiently

small:

$$\varepsilon = \min_{w \in W, n \in N} A_w(N) - A_w(N \setminus n)$$

In lines 37-38, Algorithm 4 checks whether to terminate. It terminates when a solution that includes all drop box locations has been found. Algorithm 4 returns all non-donminated solutions, and whether a solution is non-dominated by a new solution can be checked during each iteration. In an actual implementation, the order of lines within Algorithm 4 can be optimized to reduce run time (e.g., checking the condition on line 18 before the condition on line 17 may result in shorter run time).

Algorithm 4 DBLP Heuristic (ε)

input A DBLP instance defined by $(N, T, E, W, N_w, \mathbf{c}, \mathbf{v}, \mathbf{a}, q)$

(CTP)′: Find initial CTP solution when $q \geq 1$

1: $(CTP)'$:= instance to the CTP defined by $(N, T, E, W, N_w, \mathbf{c})$
2: $x' \in \{0,1\}^{|E|},\ y' \in \{0,1\}^{|N|}$:= heuristic solution to $(CTP)'$

(CTP)″: Find second CTP solution when $q = 2$

3: $W' := W \setminus \{w \in W : |\{n \in N_w : y'_n = 1\}| \geq 2\}$
4: $T' := T \cup \{n \in N : y'_n = 1\}$
5: $N'_w := N_w \setminus T'' \quad \forall\, w \in W'$
6: $(CTP)''$:= instance to the CTP defined by $(N, T', E, W', N'_w, \mathbf{c})$
7: $x'' \in \{0,1\}^{|E|},\ y'' \in \{0,1\}^{|N|}$:= heuristic solution to $(CTP)''$

Initialization

8: $N^0 := \{n \in N : y_n = 1\}$ selected locations defined by y' (when $q \leq 1$) or y'' (when $q = 2$)
9: $\mathcal{C}^0 :=$ collection tour defined by x' (when $q \leq 1$) or x'' (when $q = 2$)
10: $C := \{\mathcal{C}^0\}$ a set of previously found solutions
11: $r^1 := 0$

Iterative Improvements

12: **for** $k = 1, 2, 3, \ldots,$ until return **do**
13: $\quad (i^*, j^*) = \emptyset$
14: $\quad \theta^* = 2\pi$
15: $\quad$ **for** $i \in (N^{k-1} \setminus T)$ or i represents no drop box **do**
16: $\quad\quad$ **for** $j \in (N \setminus N^{k-1})$ or j represents no drop box $(i \neq j)$ **do**
17: $\quad\quad\quad$ **if** $|N_w \cap (N^{k-1} \cup \{j\} \setminus \{i\})| \geq q \quad \forall\, w \in W$ **then**
18: $\quad\quad\quad\quad$ **if** $r^k \leq \min_{w \in W} A_w(N^{k-1} \cup \{j\} \setminus \{i\})$ **then**
19: $\quad\quad\quad\quad\quad \Delta\hat{c}(i,j) =$ estimated change in tour cost from removing i and inserting j

20: $\Delta r(i,j) = \min_{w \in W} A_w(N^{k-1} \cup \{j\} \setminus \{i\}) - \min_{w \in W} A_w(N^{k-1})$
21: **if** $\theta(\Delta r(i,j), \Delta \hat{c}(i,j))$ computed using equation (9.1) $< \theta^*$ **then**
22: $(i^*, j^*) = (i,j)$
23: $\theta^* = \theta(\Delta r(i,j), \Delta \hat{c}(i,j))$
24: **end if**
25: **end if**
26: **end if**
27: **end for**
28: **end for**
29: $N^k := N^{k-1} \cup \{i^*\} \setminus \{j^*\}$
30: $\mathcal{C}^k :=$ heuristic tour over N^k
31: **if** $\mathcal{C}^k \in C$ **then**
32: $r^{k+1} := \min_{w \in W} A_w(N^k)$
33: **else**
34: $r^{k+1} := \min\{\min_{w \in W} A_w(N^k), r + \varepsilon\}$
35: $C = C \cup \{\mathcal{C}^k\}$
36: **end if**
37: **if** $N^k = N$ **then**
38: **return** Identify and return the non-dominated solutions in C
39: **end if**
40: **end for**

9.2 Access Function Parameters

The access function used throughout this chapter is modeled after the conditional/multinomial logit model from discrete choice theory. With a strict interpretation of the model, the access function value takes the form

$$A_w(N^*) := \frac{v_w^1 + \sum_{n \in N^*} a_{nw}}{v_w^0 + v_w^1 + \sum_{n \in N^*} a_{nw}} = \frac{e^{U_w^1} + \sum_{n \in N^*} e^{U_{wn}}}{e^{U_w^0} + e^{U_w^1} + \sum_{n \in N^*} e^{U_{wn}}}$$

where U_w^0 represents the utility of not voting, U_w^1 represents the utility of voting using the non-drop box voting system, and U_{wn} represents the utility of voting by using drop box n. The value of the access function value then represents the probability that an individual chooses to vote using any of the pathways available to them.

According to an economic theory of election participation, potential voters decide

whether to vote by comparing the cost to vote and the potential benefits from voting [56]. This idea was later codified as a linear combination of benefits and costs in the form of [161]

$$\text{Utility} = \text{Benefits} - \text{Costs}$$

McGuire et al. [122] found that a decrease of one mile to the nearest drop box increases the probability of voting by 0.64 percent. We use these ideas to justify the method by which we set the value of a_{nw} for all $n \in N$ and $w \in W$, which was presented in Section 4.5. We identify a hypothetical function of the form $a_{nw} \approx e^{\text{Benefits}-\text{Costs}}$ that aligns with the findings from McGuire et al. [122]. We find that $a_{nw} \approx e^{\text{Benefits}-\text{Costs}} = e^{2.5-D}$ where D is the distance between the voter and the drop box is an appropriate model to validate our method against. Table 9.1 is used within our assessment of this hypothetical function. The second column of Table 9.1 provides an estimate of the increase in voter turnout for a region when a single drop box a distance of D miles away is added to a voting system that currently has no drop box, assuming $2.5 - D$ is an appropriate model for the voters' utility. For example, when a drop box is located a distance of 0.2 miles from a voter, the expected increase in voter turnout is 2.7%. We assume a 70% turnout in prior elections for this region. The values in the second column are calculated as follows

$$\frac{0.7 + e^{2.5-D}}{1 + e^{2.5-D}} - \frac{0.7}{1}$$

where the distance D is given in the first column. The values in the third column represent the estimated impact of a one mile decrease to the nearest drop box. The values are calculated by taking the value in the second column for D and subtracting the value of the second column for a distance that is one mile longer $(1 + D)$. For example, the value in the first row is calculated by computing $0.027 - 0.011$, where the first value corresponds to $D = 0.2$ and the second corresponds to $D = 1.2$. The average of the values in the third

column is 0.0061. This roughly aligns with the 0.64 percent found by McGuire et al. [122] as desired.

Table 9.1: Implications of $e^{2.5-D}$ description of the a_{nw}'s assuming 70% turnout in the voting system without any drop boxes.

Distance to Drop Box (mi), D	Marginal Increase in Access Function Value	Benefit of 1 mile Decrease to Drop Box (Resulting in D)
0.2	0.027	0.017
0.4	0.023	0.014
0.6	0.019	0.012
0.8	0.016	0.010
1.0	0.013	0.008
1.2	0.011	0.007
1.4	0.009	0.005
1.6	0.007	0.005
1.8	0.006	0.004
2.0	0.005	0.003
2.2	0.004	0.003
2.4	0.003	0.002
2.6	0.003	0.002
2.8	0.002	0.001
3.0	0.002	0.001

We now validate the a_{nw} values used in Section 4.5. In Figure 9.1 we plot (•) the a_{nw} value for 1,193 randomly sampled pairs $n \in N$ and $w \in W$ against the distance, D, between n and w. Overlaid on these points (—) is the function $e^{2.5-D}$ where the cost is the distance D between n and w. We find that the proposed method from Section 4.5 produces a_{nw} values that roughly align with the function $e^{2.5-D}$. There is variance from the hypothetical line, especially with smaller distances. This is because we consider additional modes of transit and other factors in the actual calculation of a_{nw}. This is desired as it adds more realism.

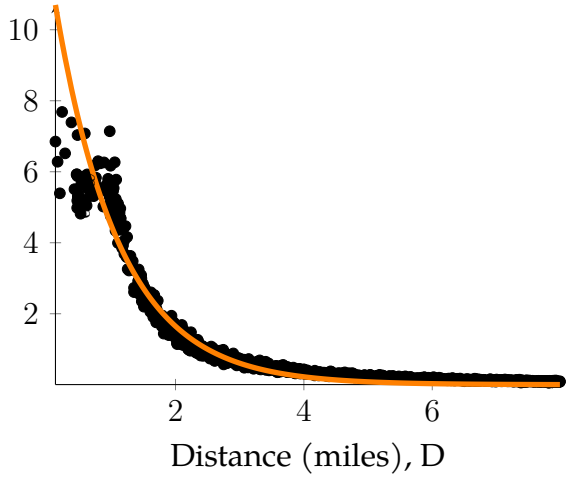


Figure 9.1: The a_{nw} value and distance between n and w for a sample of 1193 pairs (•) of $n \in N$ and $w \in W$ overlaid with the hypothetical $e^{2.5-D}$ (—) for which $D < 8$.