Optimizing Population Healthcare Resource Allocation Under Uncertainty Using Global Optimization Methods

David Linz

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

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Zelda B. Zabinsky, Chair

Archis Ghate

Paul Fishman

Program Authorized to Offer Degree:
Industrial and Systems Engineering

University of Washington

Abstract

Optimizing Population Healthcare Resource Allocation Under Uncertainty Using Global Optimization Methods

David Linz

Chair of the Supervisory Committee:
Professor Zelda B. Zabinsky
Industrial and Systems Engineering

Due to the rise in American healthcare costs, clinic administrators are increasingly concerned with optimally delivering service to patients. Due to the complex and uncertain nature of patient demand and other factors, models for healthcare systems may have to rely on discrete event simulation that incorporates random effects in order to realistically describe systems. Subsequently, optimization for decision-making must be applicable to objective functions with noise, which are often the output of complex simulation models. Second, many stakeholders may have an aversion to the risk generated by the system's uncertainty. For this reason, a well suited optimization approach must provide solutions to problems with stochastic "black-box" objective functions that provide insight to decision makers. This dissertation research has two main objectives: first, to develop models that can generate robust optimal staffing recommendations for healthcare systems in order to minimize the risk to patients while considering system constraints, and second to develop new simulation optimization theory and algorithms that can effectively minimize noisy black-box objective functions.

The first research objective is met by addressing two practical problems concerning the delivery of medical services to patients in a patient-centered medical system using a modeled decision making framework. The first problem concerns locating specialist care across geographically distributed clinics with uncertain demand. This problem highlights the trade-offs between risk and an average penalty function associated with centralized versus distributed care. The second problem addresses the question of optimal panel design in primary care that combines both operational and

strategic decisions. Since the second model cannot be easily written with closed-form equations, a discrete event simulation model is created to measure the effectiveness of chosen paneling policies in delivering care to patients.

The second research objective is met by developing two adaptive random search theoretical frameworks with provable finite time results and exploring partition-based algorithms for global optimization with noise. The two theoretical frameworks are called Quantile Adaptive Search (QAS) and Hesitant Adaptive Search with Estimation (HAS-E). Under certain assumptions the expected number of function evaluations of HAS-E and QAS increases only linearly in dimension.

This dissertation explores the implementation of partition-based algorithms that focus on sampling within quantiles to address problems with a higher number of dimensions. First, an extension to Optimal Computational Budget Allocation (OCBA) partition-based random search is developed that uses a look-ahead algorithm to improve optimizer performance. Second, an extension of the Nested Partition algorithm is adapted to sample points from a decreasing quantile level set. Third, an algorithm that samples from successive quantile level sets through the application of the Probabilistic Branch and Bound (PBnB) algorithm for level set approximation is explored. Finally, the dissertation also develops an algorithm where the PBnB algorithm is incorporated into a Nested Partition framework and the target quantile is decreased iteratively.

To provide a broad overview of potential black-box optimization for our applications, this dissertation contains research on benchmarking the numerical performance of derivative-free optimization techniques in a variety of contexts. The dissertation contains numerical results in benchmarking the effectiveness of a single observation with a "shrinking ball" approximation when estimating the objective function of a problem with noise. In addition to benchmarking existing algorithms, this effort also includes numerical performance analysis of the newly developed algorithms in this dissertation.

Overall, this research contributes to the advancement of stochastic global optimization methodology in order to practically improve real-world decision making. With the developed algorithms, healthcare administrators are able to generate near-optimal strategies for staffing and resource allocation and gain a better understanding of trade-offs in resource allocation that enable risk-averse decision makers to better serve patients.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Professor Zelda Zabinsky for her constant encouragement, guidance, and support throughout the writing of this dissertation. Professor Zabinsky provided wisdom and direction for my efforts. Her presence has been indispensable to the development of my research interests. I would also like to thank Professor Paul Fishman for serving on my thesis committee and for providing guidance in understanding healthcare decision making. Professor Fishman has been a great teacher when applying optimization tools to the problems of healthcare management. I would also like to thank Professor Archis Ghate for serving on my thesis committee and for providing advice throughout my graduate studies, as well as Professor James Burke for offering his time to serve on the committee and offering suggestions for the research problems. I would also like thank Professor Joseph Heim for his help and insight into the applied portion of my research.

I am grateful to the support provided by the faculty and staff within the Industrial Engineering Department at the University of Washington, including the Chair Professor Linda Boyle, Jennifer Wallace, Kellus Stone, and Jennifer Dutton. I would like to thank Group Health for their cooperation and support as well as the staff and administration of the University of Washington Graduate School.

I also want to express my gratitude to those who have provided spiritual support. Starting with the Almighty, my parents who raised me, my wife who provided support throughout the last year of my thesis, and my spiritual community at Blessed Sacrament Parish. In particular, I would like to dedicate this thesis to my father who first inspired me to explore science and math and who himself served as an inspiration for my graduate career. I would like to also thank those individuals who have helped me through the last five years who are unnamed here.

The research presented in this dissertation was funded in part by the National Science Foundation, NSF Grant CMMI-1632793.

Chapter 1

INTRODUCTION

1.1 Motivation

As need for effective and efficient ways of delivering healthcare increases [71], there is an increased demand for mathematical models that can provide policy recommendations to deliver quality care within the constraints of a healthcare system [34]. In particular, the organization of scarce medical staff, and the allocation of physicians time is critical to a healthcare system's ability to efficiently deliver quality of care to patients. To address this problem, mathematical models can capture key system objectives and critical trade-offs to provide insight into potential outcomes of proposed policies. Subsequently, optimization algorithms can be used in conjunction with the developed models to locate optimal policy decisions and to provide better insights into organization policy [10].

Strategic level models for allocating staff time have been shown to be effective tools in managing capacity allocation in the medical field [90]. However, due to the comparatively small amount of work done in medical staffing capacity management, there are a variety of important concerns that medical decision models often do not take into account. All feasible medical solutions must comprehend the real limitations imposed by physician preferences (given that professional preference play a major role in determining the staffing policy). Additionally, clinics need to account for how operation-level scheduling and patient-flow impacts care disruption and service [38]. Ideally a decision maker should account for patient information concerning morbidity, geographic location, and behavior (e.g., use of virtual care and proclivity to travel). Ultimately a useful capacity model requires a balance between providing consistent care, meeting physician preferences, and minimizing overall costs.

Another issue for strategic capacity management is accounting for the risk of poor outcomes generated by infrequent scenarios [58]. The problem is necessarily complicated by the multiple sources of uncertainty in medical care (patient demand, procedural uncertainty, or measurement error). This issue is especially pronounced when it comes to describing specific risks to patients or

patient populations. A decision maker can effectively use a model that recommends strategies based on constraints of location, physician preference, and patient demographics to improve resource management. Additionally, mathematical models can be used to quantify the risk of bad outcomes and assist strategic planning for risk averse decision makers.

The problem of optimal strategic capacity decisions is complicated due to the fact that the relationship between decisions and healthcare outcomes may not be easily modeled through a closed form mathematical model. Often, discrete event simulation (DES) is used to accurately model the system constraints and trade-offs [26]. Discrete event simulation can model complex systems and is useful for its ability to generate relatively accurate observations of system outcomes more cheaply than performing real-life experiments. DES models operate as "black-box" functions and can not be optimized using standard mathematical or convex optimization techniques. However, with effective black-box optimization algorithms, a DES model can provide strategies to decision makers in the context of staffing healthcare services to serve a patient population.

Partition based optimization algorithms have been effectively used for simulation applications. Furthermore, algorithms such as Probabilistic Branch and Bound [104], provide statistical guarantees of obtaining solutions within a certain quantile level set of the optimal solution. In combination with a numerical model, partition-based algorithms can isolate regions in the domain and provide insight into locating optimal and near-optimal strategies for use in healthcare staffing decision making.

This dissertation addresses the applied problem of strategically allocating medical resources and staffing hours, and the methodological problem of developing simulation-optimization algorithms for problems with a black-box objective function. A key contribution of this research develops an adaptive random search algorithm and a means of applying it to a practical problem. Additionally, our contribution includes practical benchmarking efforts that compares the relative effectiveness of optimizers when applied both to test problems and practical simulation models for policy making.

1.2 Research Contributions

The dissertation explores a general resource allocation problem under uncertainty within the context of healthcare and was motivated by two specific problems concerning resource allocation and

staffing of medical professionals. First, we examine the problem of locating specialist staff within a geographic region via a location-based model that takes into consideration risk aversion though the use of Conditional Value at Risk (CVaR). Second, we examine the problem of managing primary care patient flow with panel organization, exploring the trade-offs involved in paneling through use of a discrete event simulation. Both of these practical problems share common issues concerning risk, multiple objectives, and trade-offs between different stakeholders.

In order to obtain effective policy recommendations from the developed models, we explore new global optimization algorithms that can be applied to the problem of optimal healthcare staffing management, and solve general resource allocation problems. Here we analyze two adaptive random search frameworks for global optimization, and provide practical partition-based algorithms that quickly locate near optimal solutions for a black-box problem. After exploring the practical and theoretical dimensions of algorithms, we provide a general benchmark of optimizer performance in comparison to other global optimization algorithms.

Subsequently, the dissertation addresses these objectives in the following order:

1. Develop models that provide strategic policy recommendations for healthcare staffing and resource allocation under uncertainty and risk-aversion

    (a) Optimize a staffing model for locating physician hours across a geographically dispersed set of clinics under risk-aversion

    (b) Optimize general simulation model for patient paneling that can balance patient and physician needs

2. Develop a global optimization framework that can generate acceptable solutions to an optimization problem with noise using partition-based techniques

    (a) Develop a framework for providing finite-time analysis for a class of adaptive random search algorithms that sample from a series of nested quantile level sets

    (b) Create partition-based algorithms motivated by finite-time performance analysis

    (c) Formally benchmark the performance of existing optimization algorithms and compare to the developed algorithms to better understand numerical performance

The research in this dissertation fulfills these objectives. We address the first research objective with a closed-form problem for staffing specialists (see Chapter 3 Section 3.1), and address the paneling of patients with black-box objective function derived from a discrete-event simulation model (see Chapter 3 Section 3.2).

The first goal, $1(a)$, concerns a model for specialist staffing in a set of geographically distributed clinics extending existing research on the topic [61]. This approach focuses on the risk-aversion that a decision maker has to being understaffed, and balances that risk-aversion against an objective "penalty" concerning average travel time and general staffing expenses. The research conducted in this section largely focuses on the impacts of balancing risk and penalty when it comes to policy decisions. We demonstrate that policy outcomes differ when a decision maker is risk averse and that the solution itself is highly sensitive to heavy-tails in the demand distribution. This research has been submitted in a paper to the IIE Transactions on Healthcare Systems Engineering [58].

The goal, $1(b)$, proposes a high-fidelity discrete-event simulation (DES) to account for operational and scheduling factors concerning patient organization in primary care panels building on existing research in [7, 33]. This model accounts for populating panels, locating physicians, and setting virtual care hours such that patient wait times, no-shows, and deferments, and travel times are reduced. While the initial simulation model focuses on optimizing internal metrics, we leave future models open to using more common healthcare evaluation metrics such as *HEDIS* (Healthcare Effectiveness Data and Information Set) or *QALYS* (Quality Adjusted Life Years).

The second research objective concerns developing, analyzing, implementing, and testing optimization algorithms that can adequately provide solutions. A theoretical analysis of a optimization frameworks (see Chapter 4) and the development of new implementable algorithms (see Chapter 5), with benchmarking results (see Chapter 6) fulfills the second research objective. The general strategy is use global optimization methods to approximate a near optimal solution to programs with stochastic objective functions. To this end, the dissertation fulfills this objective in three parts: $2(a)$ developing a theoretical adaptive search framework with provable finite-time results; $2(b)$ developing partition-based optimization algorithms with hopes of approximating the ideal performance of the theoretical framework; and $2(c)$ benchmarking existing and newly developed optimization algorithms for purposes of understanding numerical performance.

To address the goal $2(a)$, two theoretical frameworks are presented that allow for finite-time

analysis of algorithm performance. The first algorithm, Hesitant Adaptive Search with Estimation (HAS-E), extends existing adaptive random search research concerning Hesitant Adaptive Search to problems with noise and estimation [17, 103, 105]. We prove that, under certain circumstances, the expected number of iterations needed to obtain a value within a fixed range of the optimum increases linearly in dimension, and the number of replications needed to obtain a value within a fixed range of the optimum increases as a cubic function of the domain dimension. The second algorithm, Quantile Adaptive Search, extends the results of adaptive random search algorithm to a framework that samples from a series of nested quantile level sets. We demonstrate that under certain conditions, the number of iterations the algorithm requires to obtain a value within a defined range of the optimum increases linearly in dimension.

Starting with finite-time analysis provided by adaptive random search techniques, the next contribution addresses goal $2(b)$ by implementing an algorithm that successively samples from a set of decreasing quantile level sets using partition-based algorithms. We explore four practical algorithms that use partition-based methods to sample better points for a black-box function. The first algorithm discussed uses a partition-based random search with a "look ahead" methodology in order to optimally allocate a computational budget (OCBA) [54]. The second algorithm extends the Nested Partition framework [95, 96] to sampling from a quantile level set [55] with provable asymptotic results. Lastly, we explore two different modifications of the Probabilistic Branch and Bound (PBnB) algorithm for level set approximation [45, 104] in order to sample from a series of nested levels sets. The first algorithm successively lowers the target quantile level to target more specific level sets, the second fits the PBnB algorithm into the nested partitions framework, maintaining one promising region and using the PBnB algorithm to determine the most preferable region to branch.

The goal $2(c)$ involves the benchmarking of global optimization algorithms. Starting with a previous benchmark study done on common test problems, we explore the result of noise and dimensionality in optimizer performance [59]. Furthermore, we go on to compare developed optimization implementation of new algorithms to classic optimization algorithms over a variety of test problems.

The dissertation is structured as follows. Chapter 2 consists of a literature review outlining decision support tools for healthcare applications and reviewing some methods for optimally locating staffing resources. The literature review also summarizes various simulation optimization methodologies used to solve black-box stochastic optimization problems and then discusses the ex-

isting research on adaptive random search. In Chapter 3, the dissertation outlines work done on the subject of risk-sensitive stochastic optimization for healthcare applications, detailing a geographic model for specialist staffing and then a simulation model for locating staffing along with the solution methods used to generate their solutions, fulfilling research objective 1. In Chapter 4, we discuss the theoretical contributions to developing adaptive search algorithms for global optimization. Further methodological contributions involving implementations are discussed in Chapter 5. Finally, we explore the general performance of our developed optimization algorithms through a benchmarking effort in Chapter 6, fulfilling research objective 2. The dissertation concludes in Chapter 7 with a summary of results and a discussion of future research.

Chapter 2

LITERATURE REVIEW

2.1 Optimization and Healthcare Staffing

It is widely accepted that healthcare costs are increasing significantly. Statistics show that 15.2% of the U.S. *Gross Domestic Project* is dedicated to healthcare spending. Moreover, personal healthcare costs have been increasing significantly [71, 77]. To address concerns with rising costs, interest has arisen in improving decision making and general system management in healthcare. Furthermore, delivering affordable quality care through proper resource allocation is of increasing interest [24, 80]. More specifically, strategic staffing [98], scheduling, and patient flow management is critical to the overall efficient delivery of healthcare to patients [84].

A key component in delivering patient-centered care is optimizing staffing capacity. Some evidence demonstrates that new strategies could handle higher levels of patient demand while improving the provision of care [33, 101]. The proper sizing, composition, and assignment of panels directly impacts the level of timely treatment and continuity of care [32, 99]. Due to the connection to patient healthcare outcomes, a particular interest has been increasing continuity and timely access through optimal staffing strategies [22, 30, 70, 75, 89]. Currently, there is a broad literature focusing on staff-based research allocation both from a management side [35, 39, 68, 70, 69, 72, 90, 97] and optimization perspective [7, 8, 34, 78, 86, 90, 99, 109]. Some research has also been done on geographic staff location [14, 61]. For the most part, models have employed either stochastic numerical models or queuing models to make decision recommendations. So far, simulation optimization has seen some use in staffing applications [9, 109]. Given the diversity of rules and decisions that need to be supported, simulation optimization is a fruitful approach to the problem of making risk-averse strategic staffing decisions.

Simulation models are useful due to their ability to model a variety of different scenarios and a variety of different outcome metrics. The use of simulation already has widespread application in other areas of healthcare decision making [11, 37]. Recently, some attention has been paid to the

subject of simulation models in out-patient services, clinic planning, emergency departments, and hospitals [26, 36, 49]. Generally simulation modeling has been useful in determining patient flow [36, 62] and capturing the effects of complex series of rules more effectively than relying on simplified numerical models or direct experimental measurements. Simulation models have also been effective at modeling general wait time reduction [100] and effectively supporting staff management in long term care [1, 2, 109]. Since specific risks to patients often arise from acute occurrences which are hard to capture in direct numerical models, a simulation model might logically be expanded to this application especially when the focus is on patient outcomes and not aggregated system metrics.

An increasingly common approach to measuring healthcare performance is "patient-centeredness" (the degree to which a healthcare system is organized to optimize the delivery of services to patients). This management perspective focuses the provision of resources to optimize results relative to patient perspectives. The impact of continuity of care and access to care is widely understood to be a priority for patients [29, 30]. Moreover, risk-aversion can also be observed in patient populations [48, 63] and some operations research for managing a kidney exchange system have been explored [110]. From the patient-centered perspective, patients' risk-aversion is a key concern. Moreover, a successful patient-centered model would account for different levels of risk aversions more pronounced among certain demographic groups [27]. Therefore it is reasonable to consider a risk-sensitive approach to accurately account for patient priorities in some scenarios.

However, several key concerns are not fully addressed in existing literature. First, the impact of staffing physicians relative to geographic location is not directly addressed in most existing staffing models. Second, none of the strategic decision support tools developed to this point have been focused on patient risk and different levels of aversion patient populations might have towards excessive wait and travel time [61]. Furthermore, the introduction of virtual care (e.g., e-mail and direct messaging with physicians) as a large part of most modern primary or specialists care services has not been incorporated into staffing models, which might help healthcare systems take into consideration the placement of staffing in a system of geographically distributed clinics. Given the trade-offs between centralized and distributed care, combined with the asymmetrical use of virtual care across populations, there is a need for policy recommendations that could focus on balancing patient and physician needs when developing staffing policy for both locating medical staff geographically and directing patients to receive care. We extend this work in Chapter 3, first by examining the opti-

mization of risk-averse specialist staffing in a set of geographically distributed clinics (Section 3.1), second by modeling and optimizing panel design and location using a simulation and simulation optimization algorithms (Section 3.2).

2.2 Simulation Optimization Algorithms

In recent years, the field of stochastic global optimization has been steadily developing [28]. In particular, optimization of stochastic systems with no closed form expression has been of particular interest. These methods, also referred to as "simulation optimization," develop optimal or near optimal solutions based on statistical samples observed from the function in question. Since analytical properties of the function cannot generally be obtained, many of these methods rely on Monte Carlo approaches for optimization or other types of random search.

A large body of research exists describing black-box global optimization. An excellent review of stochastic optimization methods can be found in [43] that describes many modern approaches to stochastic optimization using Monte Carlo methods. Methods typically fall in several categories. First, there are classic methods such as sample average approximation (*SAA*) and sequential optimization methods called stochastic approximation (*SA*) [53]. Other methods include meta-model methods, simulated annealing [4] and kriging [79], genetic algorithms [16], cross-entropy [88], and derivative free non-linear programming [12, 52].

Of particular interest to this research are algorithms that iteratively partition the domain, also called partition-based methods. Partition-based optimization methods have seen success in the field of simulation optimization. One common method is Stochastic Branch and Bound [76] which employs statistical estimators to develop upper and lower bounds for the expected value inside a subregion. Additionally, Nested Partition methods [94, 96], update a fixed set of nested partitions in order to improve the probability of sampling an optimal solution. More recently, the Probabilistic Branch and Bound method has been developed which employs order statistics to obtain bounds on the objective function [44, 83].

Determining the effectiveness of different optimizers can also be quite challenging. Some papers have benchmarked the effectiveness of optimization algorithms for black-box functions on discrete or continuous domains [28, 60, 73], including applications such as [3, 74, 81, 85]. The

benchmarking papers have surveyed a diverse set of different optimization algorithms. However, much optimization research does not directly address the issue of different methods estimating an objective value with noise. Here, there is room both for expanding benchmarking of algorithms in the context of real-world optimization problems and for expanding benchmarking efforts to explore different mechanisms for accounting for noise and estimation.

Another desirable quality of some optimization algorithms is provable performance results. So far, very few algorithms have provable results in finite time. However, one set of theoretical algorithms, adaptive random search algorithms [105], possesses sound theoretical properties under certain conditions.

The most basic adaptive random search is Pure Adaptive Search (PAS). The PAS algorithm samples from an improving set of points at each iteration. Finite-time analysis proves that, under certain conditions, the expected number of iterations until PAS samples within a range of the minimum value increases only linearly in terms of the dimension of the domain [107]. These finite time results have also been extended to optimization problems over discrete domains [108]. However, the requirement of improving at every iteration makes the algorithm difficult to implement practically. The PAS algorithm is listed in Appendix A.

Another adaptive random search algorithm, called Hesitant Adaptive Search (HAS), generalizes PAS, allowing for a "hesitation" (where the algorithm does not improve) within a certain specified probability that depends on the best objective function value sampled. Finite-time analysis of hesitant adaptive search allows for a closed form expression for the expected number of iterations until reaching a specified value above the minimum function value [102, 103]. The HAS algorithm is listed in Chapter 4.

Further development of adaptive random search algorithms include the Annealing Adaptive Search (AAS), which is based on sampling from a Boltzmann distribution [87]. Annealing Adaptive Search shares many of the good finite time results of Hesitant Adaptive Search (under certain conditions). Furthermore, the use of the Hesitant Adaptive Search has been used to analytically derive a cooling schedule for simulated annealing algorithms [87, 93]. Annealing Adaptive Search method, while not directly implementable, can be approximated through hit-and-run Monte Carlo methods which allow an optimization algorithm to approximately sample from a Boltzmann distribution as required. Still, adaptive random search algorithms do not directly address the question of estimation.

To address this need, the dissertation further develops algorithms for simulation optimization. First, in Chapter 4, we extend the set of adaptive random search algorithms by introducing Hesitant Adaptive Search with Estimation (HAS-E) in Section 4.1, and Quantile Adaptive Search (QAS), in Section 4.2 and show finite-time results that lead to linearity in performance under certain conditions. Second, in Chapter 5 we discuss various implementations of partition-based algorithms that attempt to handle high dimensional problems and approximately implement QAS.

Chapter 3

DECISION SUPPORT MODELS FOR HEALTHCARE STAFFING

This dissertation addresses several mathematical models that describe strategic staffing decisions within a set of geographically distributed healthcare clinics. In particular, we address the question of properly locating healthcare professionals and assigning patients for care. Starting with this general motivation, the dissertation overviews several models where staffing decisions can be linked to operational outcomes as shown in the general model in Figure 3.1.

The general model is one that links stochastic events and patient behavior with staffing decisions and healthcare results. Starting with a patient population and domain knowledge of a healthcare system, the goal of the general model is to generate a policy recommendation that will optimize a number of measurable objectives. As shown in Figure 3.1, the general model relates patient inputs, system parameters, and strategic decisions to healthcare outcomes. This model may be mathematical, statistical, simulation-based, or some combination of the two. Furthermore, due to the stochastic nature of the patient behavior, the model will take into consideration uncertainty and stochastic variables. The model's uncertainty might derive from unknown qualities about the healthcare system or subject population (e.g., morbidity and geographical distribution), future patient behavior (e.g., patient demand), or physician behavior (i.e., preferences, availability).

From a patient-centered perspective, managers are typically interested in ultimate healthcare outcomes for patients and, with enough domain-expert input, a model can relate strategic and operational decisions to their effect on patient health. Although a more sophisticated simulation model would be able to account for ultimate impacts on healthcare outcomes, in the absence of extensive domain knowledge that would allow for a strong connection between operational outcomes and healthcare, this dissertation focuses on a model that relates decisions to *operational* outcomes.

To address a strategic question of resource allocation, a model recommends a set of decisions in order to deliver a balance of desired metrics. This might be some mix of physician requirements and patient outcomes measured either by expected value, risk measure, or a linear combination between

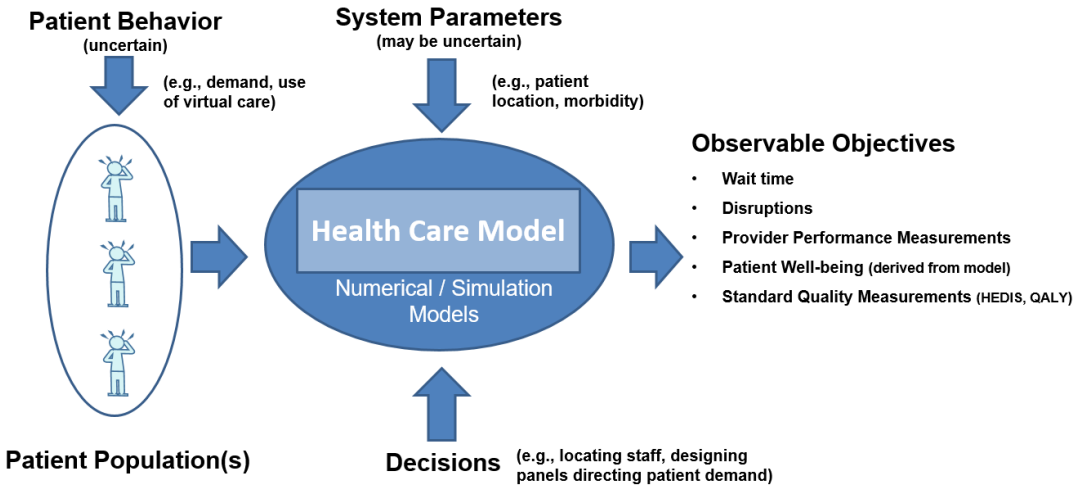

Figure 3.1: A basic decision model for staffing inside a healthcare system. The model inputs general information concerning patient behavior, demographics, system parameters, and policy decisions. The model outputs measurable outcomes for patients which might include wait time, care disruptions, and other performance measurements.

the two. With the generated solution (or set of solutions), a decision maker can better understand the trade-offs and develop strategies feasible for the system requirements while taking into account the needs of potentially risk-averse decision makers.

In this chapter, two health models are presented to deal with the problem of optimal staffing in different contexts. The first (described in Section 3.1) addresses research objective 1(a) and is a strategic staffing model that helps locate specialist hours across a geographically distributed number of locations. The work has been presented at the INFORMS Healthcare conference (Nashville, July 2015) [57] and a paper presenting the main results has been submitted for publication to the *Journal of IIE Transactions on Healthcare Systems Engineering* [58]. We consider this contribution a completion of research objective 1(a).

The second model (described in Section 3.2) provides optimal primary care panel composition for a primary care panel redesign and directly addresses research objective 1(b). Here a discrete-event simulation is used to capture specific patient and physician behavior. The model attempts to reduce metrics concerning care disruption (balks and no-shows). Three simulation optimization

algorithms are used to find optimal policy solutions. Preliminary results were presented at the 2015 INFORMS annual meeting [56]. The produced results are able to provide insights into designing and sizing panels in order to best serve a group of patients of different morbidity level and geographic regions. We consider this contribution a completion of research objective 1(b).

Given the recommended policies generated by selected algorithms, we are able to speculate about the efficiency of the optimization methods when optimizing healthcare staffing simulations. Future research will look into the possibility of expanding the number of dimensions for the simulation to capture a larger system and the effectiveness of the chosen algorithms in finding an improved policy solution in those higher dimensional spaces.

3.1 Geographical Staffing to Reduce Risk of Demand Shortfall

A critical problem in healthcare decision making is the geographic distribution of specialist care. For any geographically distributed population of patients, specialists need to be located at specific clinics in order to serve that populations' demand for care [61, 98]. To properly allocate resources, a decision maker must determine staffing levels relative to concerns about patient travel time, accessibility, and over-all costs. Generally, the goal of a manager is to staff specialists to minimize the travel time of the patients, the risk of a patient being "turned away" for a given time frame, and the general expenses of supporting specialist care.

At a strategic level, the problem could be seen as forming "clusters" of clinics that redirect patients to centralized locations where specialists are located. This problem can be conceptualized as picking preexisting clinics and redirecting populations from other clinics in the cluster to these centers for their specialist needs (an example of Group Health is shown in Figure 3.2). In terms of overall strategy this is a case of matching demand for physician-hours from the population to staffed hours in the selected clinic clusters.

As noted in Chapter 2, we are particularly concerned about the risk of "shortfall" between the patient demand for specialist attention and providers' capacity, since this relates to times when patients are turned away or experience extended delays, which are outcomes to which patients and decision makers are particularly risk averse. This contrasts the more regular costs associated with paying for staff and regular travel time.

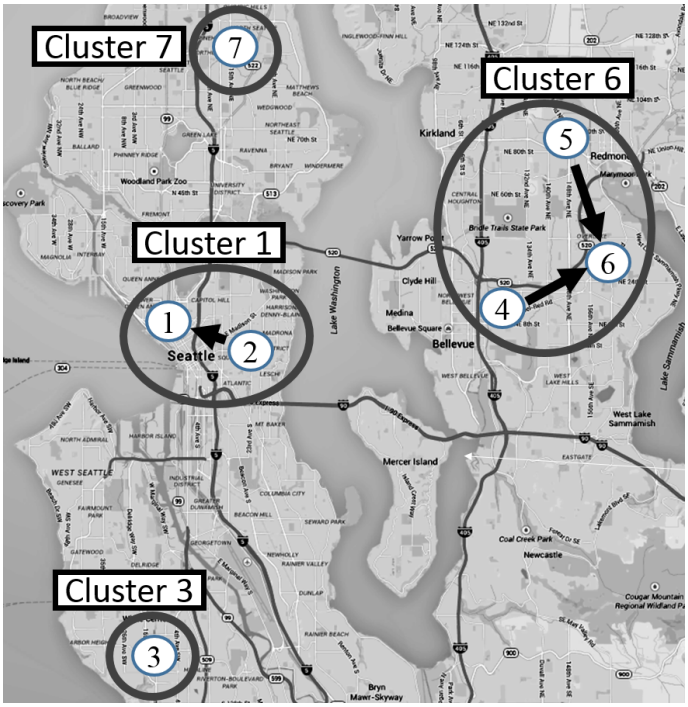

Figure 3.2: An example of seven clinics in the greater Seattle area, marked with light circles and labeled 1-7. Clusters where specialist care is shared between clinics are marked with the shaded circles. The arrows indicate the central clinic inside the cluster that provides care.

To confront these particular challenges, we propose a multi-objective optimization model that balances a risk measure (for being understaffed) with the expected loss of a penalty measure which comprises a linear combination of regular costs.

Minimize :

1. Risk of being understaffed

2. Penalty due to:

   - Staffing cost
   - Patient travel time
   - Patient unfamiliarity

Our paper [58] determines a linear expression for the number of hours understaffed at each clinic as well as linear expressions for the costs imposed by travel distance, additional staffing hours, and piece-wise linear function expressing the costs of "unfamiliarity" in larger clinics. Each equation is written in terms of demand random variables localized at clinics creating a stochastic expression for the staffing costs, number of hours over-staffed, and patient travel time.

The critical issue in the staffing model is developing a trade-off between the risk of understaffed hours and the average penalty. For this purpose, a risk measurement is used to characterize the excess of demand relative to staffed specialist hours.

The model uses a risk measurement of the hours understaffed for one objective function and contrasts it against the expected value of a penalty which is a weighed linear combination staffing costs, patient travel time, and patient unfamiliarity (calling these "risk" and "penalty" respectively). Combined with capacity constraints and demand satisfaction constraints, this forms a risk-based multiple-objective program.

To solve the risk-based optimization problem, the paper makes use of Conditional Value at Risk (CVaR). The formula for Conditional Value of Risk allows the risk portion of the objective function to be replaced by linear approximation and the optimization problem can be solved using mixed-integer linear-programming techniques. An efficient frontier can then be plotted as shown in either of the graphs in Figure 3.3.

A key contribution for our paper is understanding the importance of using a risk measurement for demand shortfall as opposed to simply relying on an average measurement of demand shortfall. The risk measurement emphasizes the losses that come from more infrequent surges in demand, a critical concern for healthcare managers. This is especially important when looking at heavy-tailed distributions where large demands comprise a significant portion of the demand distribution. Use of the Conditional Value at Risk allows these effects to be captured in the proposed model for general demand distributions, a feature that distinguishes our work from previous approaches to the problem [61].

From available proxy-data we observe a significant non-normal and heavy-tail behavior in patient demand for specialist care. Furthermore, we compare the optimal decisions made for sample data taken from a normal distribution and Weibul distribution with a heavy tail when optimizing a risk-averse objective function. Both of these efficient frontiers are graphed in Figure 3.3. Based

on the case study, we see that the two scenarios have significantly different optimal points with respect to the risk-penalty trade off, and that points at the same penalty limit recommend significantly different staffing strategies.

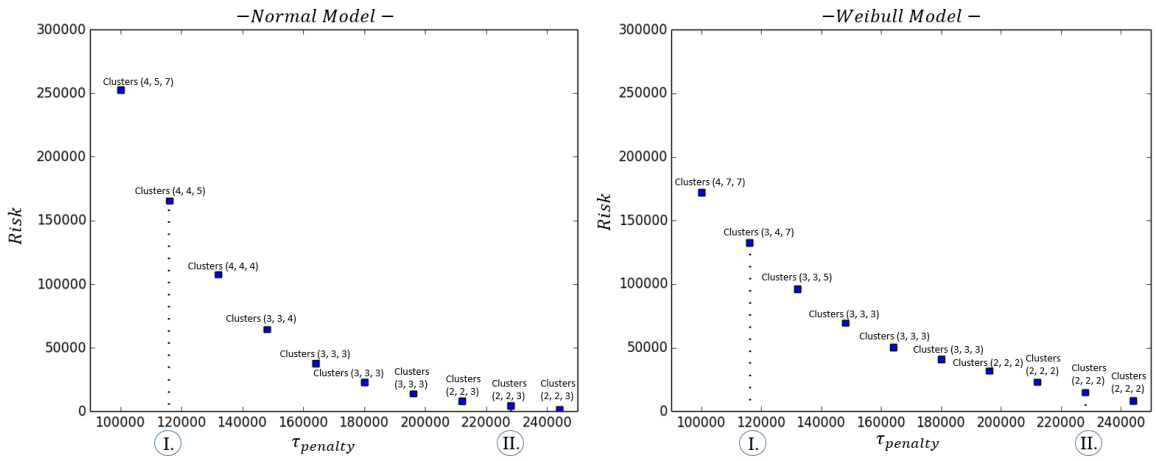

Figure 3.3: The constructed efficient frontier (non-dominated solutions) for the optimization model where demand is modeled with a normal distribution (left panel) and a Weibull distribution (right panel). Each data point is labeled with the number of clusters for each specialist type in the final optimized solution (in order: oncology, endocrinology, and behavioral care).

This graph demonstrates the practical problem with modeling a risk-averse objective without accurately accounting for demand distribution's tail effects. Without observing these effects, the program results in very different strategic recommendations, in this case creating more clusters (less aggregation) than when the demand is assumed to be normal. The contract between the two models is compared in Figure 3.4 where the location of clusters is illustrated for scenario I for both models. Details are provided in [58].

The results of this research make the following contributions to research objective 1(a). First, the model highlights the risk-averse problem of staffing specialists geographically. Second, its solution allows us a flexibility in handling different demand distributions which might be non-normal. Third, it provides a practical method through which a decision maker could be risk-averse when it comes to being understaffed relative to patient demand for care.

However, the model does have some disadvantages that might be addressed in future research. On the practical level, the linearized formulations for the objectives are based on a simplified un-

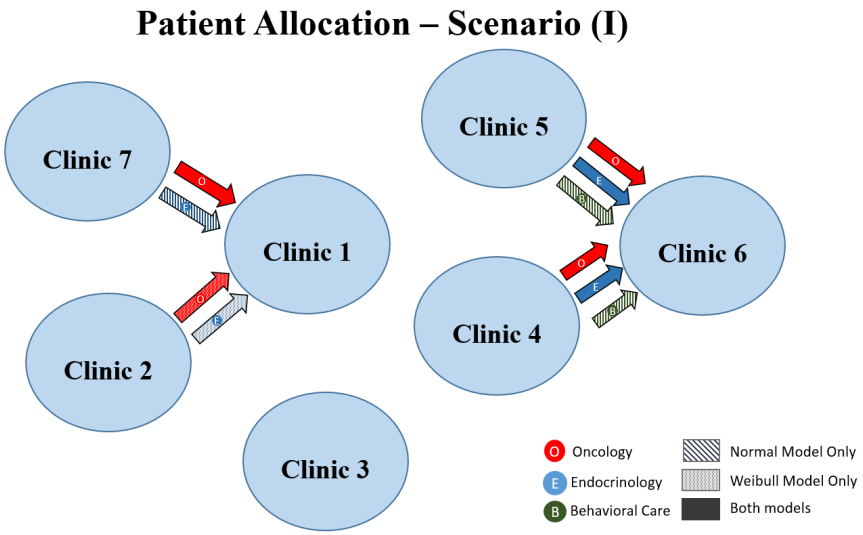

Figure 3.4: An example of staffing policy recommendation with penalty level at $\tau_{penalty} = 116000$. The lined-arrows show re-direction of patients under the normal model, the dotted arrows show the redirection of patients recommended by the Weibull model, and the solid colors show the re-direction under both models. Colors and labels distinguish the different specialist types being directed.

derstanding of how demand shortfall of staffed physician hours impacts patients. For instance, it is well understood that increasing travel time also makes it more likely that patients miss appointments, regardless of the amount of specialists staffed. These complex random relationships while easy to model via simulation are not included in the linear model. Further research described in the next section addresses optimal primary care paneling using discrete-event simulation.

3.2 Optimal Primary Care Paneling for Patient and Physician Needs

Increasingly, demand for primary care exceeds the availability of physicians able to see patients. Healthcare managers are under pressure to manage this primary care shortfall with strategic panel policy that divides a group of identifiable patient populations between physicians to balance the workload and improve continuity of service. To account for distributing care, each primary care physician has a panel of patients that look to them as the first point of contact for medical care. While physicians have discretion in panel formation, panel redesign is sometimes possible to better balance the demand between a set of physicians. Generally, better balancing patients between physician-

panels may result in better utilized physician time and more efficient provision of primary care to patients.

The question of optimal panel size and composition is sensitive to uncertainty in patient demand (which can be highly variable) as well as patient behavior and physician availability. If staffing arrangements are not optimal, clinics risk over-staffing at low demand periods, whereas high demand periods may result in delays in care, overburdened healthcare workers, and poor healthcare outcomes [98]. Clinics face the challenge of paying for excess capacity or having unmet patient demand for care. This section addresses the design of primary patient panels (relative to preexisting geographic and demographic groups), as well as the location of physicians throughout a geographically distributed network of clinics.

Model Description

Our model for primary care design considers several key factors including the amount of services demanded, the relative supply of physicians (and hours worked), geographic limitations of travel, and the need for virtual care (such as physician consultation done through e-mail and internal messaging). Since there is no consensus within either the operations research or healthcare communities on how to optimally design primary care panels to achieve the best outcomes with these considerations [84, 98], our model provides a comprehensive simulation model for panel design that takes into account the impact of demand variability, geographic location, and physician behavior on operational outcomes. Without taking these factors into account, strategies may recommend policies that result in increased deferment of care due to mis-allocated physician time or excessive travel-time on the part of patients [15].

To build and validate our model relative to real-world considerations in a local context, we parameterize our model with data gathered from the Group Health Research Cooperative in Washington State. The data includes electronic medical records concerning patient appointment frequency, demographics, and location, as well as the behavior and constraints on physicians. Based on generated samples, optimal recommendations are provided that are targeted at using patient panel re-design to obtain patient-centered outcomes.

The model describes patient behavior by modeling the use of primary care, no-shows, deferments, and virtual care using random variables. The model accounts for random variables describ-

ing physician availability as well as the randomness in demand from patients of different morbidity types. Modeling this randomness allows a more robust set of paneling recommendations that account for the uncertainty generated by patient behavior.

The model directly addresses decisions concerning panel design in the context of a geographically distributed set of clinics. The central decisions that need to be made include:

1. The composition of panels in terms of patients from different locations and morbidity levels

2. The location of panels at geographically distributed clinics

3. The allocated amount of time spent on virtual care for each physician

These decision variables directly address the need for patients to have a primary care physician that is available, close, and able to assist their needs in the proper medium (virtual or face-to-face). Given that patient needs are highly variable, these considerations need to be accounted for in a policy decision.

The key objectives of our model are: reducing the amount of patient balks, no shows, wait time, travel time, and improving physician utilization inside of a given system of clinics which serve a set of geographically distributed patient populations. We define a "balk" as anytime a patient requests a primary care appointment and does not receive care due to long wait time at the clinic. We define a "no show" as anytime a scheduled patient does not show up to a scheduled appointment. Similarly we define "wait time" as the total time a patient spends between request and fulfillment, and "travel distance" as the total distance traveled (regardless of whether they receive care or not). Determining each of these objectives involves modeling patient demand and physician availability across a number of primary care clinics using a discrete-event simulation which we formulate and describe in the next section.

Model Formulation

The model describes a set of primary care clinics, denoted $l \in \{1,\ldots,L\}$. Patients are categorized with $m \in \{1,\ldots,M\}$ patient morbidity categories and regions denoted $r \in \{1,\ldots,R\}$. The number of patient populations indexed by $i \in \{1,\ldots,I\}$ (where $I = R \times M$). We consider primary

care physicians with associated panels, denoted $j \in \{1,\ldots,J\}$ (panels correspond one-to-one with primary care physicians) which can be located at the various clinics. The general outline of the indices is shown in Figure 3.5.

For the general model formulation, a policy maker chooses physician-panel composition for each patient morbidity type in each region. The decision variables include the assignment of a certain percentage of patients in population i to a given panel j, $X_{i,j}$, and the location of the physician-panel j , Y_j which controls the location of a panel at a given clinic (e.g., $Y_1 = 2$ indicates that panel $j = 1$ is located at clinic $l = 2$). Finally, we track the use of virtual care including the amount of care time reserved for virtual interactions V_j at each of the defined panels j. The decision variables are outlined in Figure 3.6 with full specification of the variables in Table C.6.

The optimization program expressed therefore contains $(I \times J) + J + J$ decision variables. However, this can provide some difficulty for even a moderate number of panels and patient populations. For instance, using three regions and two morbidity categories (six total patient population types) with five panels results in a problem size of forty dimensions ($3 \times 2 \times 5 + 5 + 5$), which is very large for optimization. This may make the dimensionality of the problem so large as to be intractable by optimizers designed to optimize simulations as black-box functions.

To reduce the dimensionality, we separate morbidity from region and reduce the index describing patient populations i. Based on selected morbidity categories, the simulation assigns a region randomly (based on demographics) at runtime. Furthermore, specific policy recommendations are derived based on similar demographic analysis that associates morbidity categories to geographic regions. We describe this further in the description of the simulation logic in the following section.

Simulation Logic

The simulation model uses a discrete-event framework to track patient interaction with the primary care system. For each created patient entity, the model tracks patients as they arrive in the system, request care, get care from a doctor associated with their assigned panel (either virtual or face-to-face), and then records whether they experienced a “balk” or a “no-show”, along with their recorded wait and travel time.

The simulation model inputs a number of parameters that characterize system behavior. These include the relative distances between the region r and the clinics l, $d_{r,l}$, the base likelihood that

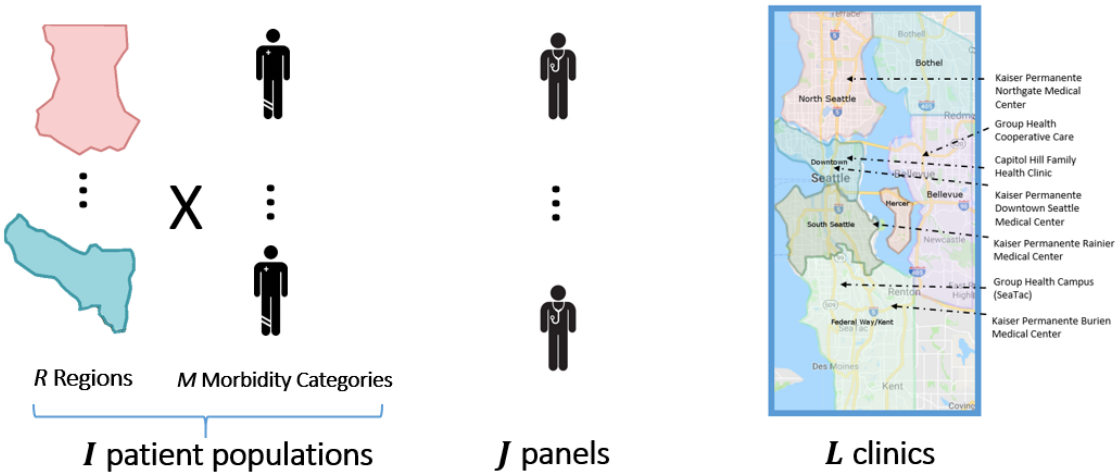

Figure 3.5: The indices as defined in the simulation model. The number of patient populations, I, split across J physician-panels, each assigned to one of L clinics. Patient populations are directly associated with a morbidity and location region by definition or by an observed statistical relationship.

a patient from population i uses virtual care, and v_i, the base likelihood that a patient will request an appointment, o_i, the likelihood that a patient of morbidity type m from region r receiving care at panel l is a "no-show", $n_{m,r,l}$, and the base wait time in the queue before a patient balks, b_i. The upper bounds of the availability of physicians j is denoted h_j. Whereas the upper and lower bound for virtual availability are v_j^{low} and v_j^{up}. Each physician panel also has a desired utilization f_j. Additionally the model inputs a certain scheduling time frame limit, $t_{frameLimits}$. The full description of model parameters and decision variables are laid out in Table C.6.

The model for patient and physician behavior is constructed as a discrete-event simulation. For each patient population i, we model the arrival of patients as a Poisson Process with an average inter-arrival time with mean λ_i. The simulation processes patients by generating a certain number of random entities (approximately 3000 with the case study) for each patient type i. The patients are assigned to a panel j randomly based on the percentage $X_{i,j}$ which in turn comes with a location defined by the clinic assignment, Y_j as denoted in Figure 3.6.

Each patient entity of morbidity m is assigned to a region r based on a statistical distribution,

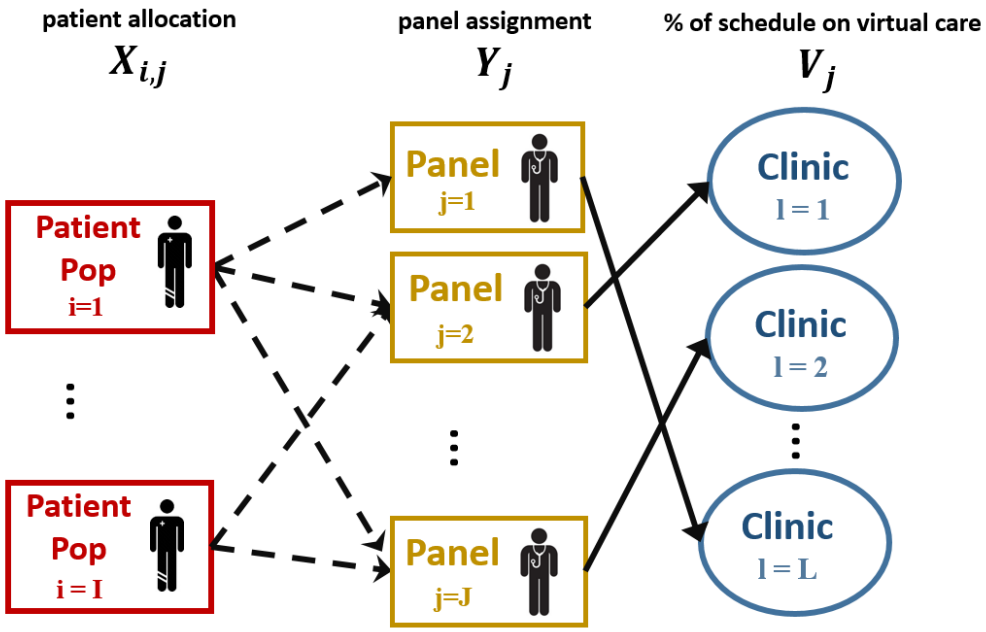

Figure 3.6: The patient redesign conducted over a geographically distributed set of clinics where patients can be redirected in between panels ($X_{i,j}$) and panels can be moved in between clinics (Y_j) and each physician-panel can have a designated time reserved for virtual care (V_j).

$\chi_{m,r}$. This assigns each patient entity to a population i implicitly. Patients from population i are either designated as requesting virtual care with probability ν_i or requesting an appointment with $1-\nu_i$. Virtual patients immediately wait for care without scheduling. Non-virtual entities are then either assigned to a first-come-first slot with probability o_i or are marked as "scheduled" and delay for a random time between $0-3$ days. To account for scheduling negotiation, patient entities get additional delay time if another scheduled patient is already receiving care when the scheduling delay has elapsed. The utilization is measured and compared to a desired utilization f_j.

Scheduled patients may experience a no-show with probability $n_{m,r,l}$ based on their morbidity and the distance between their associated region r and the clinic l. Otherwise, both first-come patients and scheduled patients accrue a travel penalty, and arrive at the clinic (after schedule delay) to wait to receive care from the physician associated with their assigned panel. If a patient's wait does not exceed a "balk time" b_i, the entity receives care after the waiting period (wait time only accounting for time waiting in clinics and not scheduling delays or time waiting for virtual care).

After leaving the system, the patient entity records whether it has balked, B_i , no-showed, N_i, or received care and record their total wait-time (virtual or non-virtual), W_i, and the travel penalty, T_i.

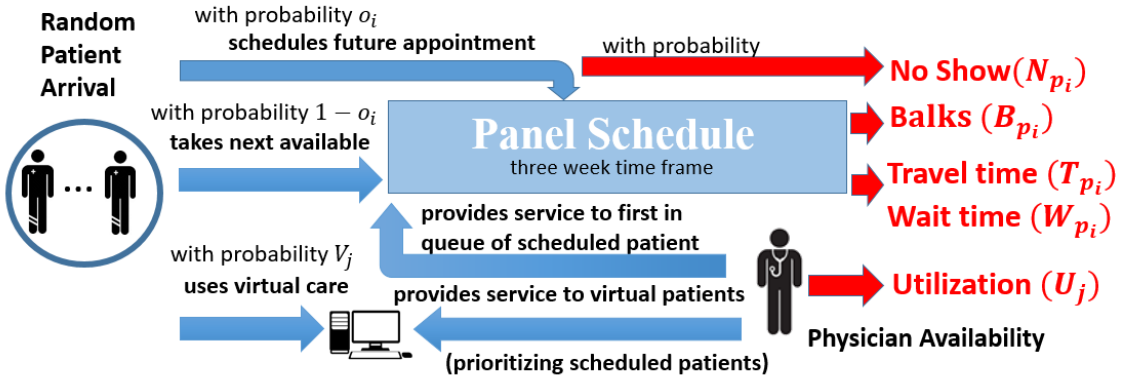

Figure 3.7: The logic governing patient scheduling, virtual care, balks, and no-shows inside a primary care panel. A patient population i moves through the system with various probabilities of experiencing a "no-show" or "balking". Physicians divide time between assisting virtual care patients and appointments. System records the number of "No Shows", "Balks", patient "Travel Time", patient "Wait time", and physician "Utilization".

Primary care physicians are tracked in the system by resource objects which become scheduled based on timed variable that is set to "virtual", "non-virtual", or "off". When in non-virtual mode, the physicians serve patients as they arrive (prioritizing the scheduled patients). During designated "virtual" times physicians only serve patients seeking virtual care at a lower service rate. The panel schedule cycles so that a physician's assigned virtual time (V_j) and total time-available to non-virtual patients equals the total availability. The recorded number of utilized physician hours are recorded as U_j. The full simulation logic is outlined in Figure 3.7.

Optimization Formulation

Based on the outputs generated by the DES model, we formulate the model as an optimization problem based on the decision variables, $X_{i,j}, Y_j, V_j$. The objective function written in (3.1) is the sum of all of the balks, wait time, no-shows, travel time, and the deviation of the utilized hours from the specific doctor's preferred time utilization, combined linearly with objective function "weights", ω_i^{balk}, $\omega_i^{waittime}$, ω_i^{noshow}, ω_i^{travel}, ω_i^{util} into a single objective function. The counters B_i, W_i, N_i,

T_i are related to the decision variables through simulation and scheduling logic as shown in (3.2). Additionally we include constraints on the number of virtual hours,, along with specific bounds for assignment ratios and values.

$$\begin{aligned} &\text{Minimize:} \\ &E\left(\sum_{i=1}^{I}\omega_i^{balk}\cdot B_i+\sum_{i=1}^{I}\omega_i^{waittime}\cdot W_i\right. \\ &\left.+\sum_{i=1}^{I}\omega_i^{noshow}\cdot N_i+\sum_{i=1}^{I}\omega_i^{travel}\cdot T_i+\sum_{j}^{J}\omega_j^{util}\cdot|U_j-f_j|\right) \end{aligned} \tag{3.1}$$

where B_i, W_i, N_i, T_i, and U_i are output from the simulation model for inputs, $X_{i,j}, Y_j, V_j$, i.e.,

$$\begin{aligned} &Sim(X_{i,j}, Y_j, V_j \text{ for } i \in \{1,\ldots,I\}, j \in \{1,\ldots,J\}) \\ &= (B_i, W_i, N_i, T_i, U_j \text{ for } i \in \{1,\ldots,I\}, j \in \{1,\ldots,J\}) \end{aligned} \tag{3.2}$$

subject to:

$$v_j^{low} \le V_j \le v_j^{up} \quad \forall j \qquad (\textit{Constraints on time physicians spend on virtual care})$$

$$\sum_{i=1}^{I} X_{i,j} = 1 \quad \forall j \qquad (\textit{All patients in population allocated})$$

$$Y_j \in \{1,\ldots,L\} \quad \forall j \qquad (\textit{Ensure only one clinic location for physician panel})$$

$$0 \le X_{i,j} \le 1 \quad \forall j \qquad (\textit{Constraints on population assignment})$$

With this formulation, the relationship between the decision variables and the component objective outputs are modeled through a discrete-event simulation. The optimization algorithm applied to this model must handle a non-linear, noisy, black-box function with mixed continuous and integer

variables. To solve this program, we appeal to simulation optimization techniques that can handle a large number of dimensions and identify promising regions of the domain for policy considerations.

Case Study

Based on the general model framework, we are able to create a specific model for purposes of re-designing a set of physician-panels across a small network of healthcare clinics within a geographic region. For purposes of our case study, we examine three clinics ($L = 3$) and five panels ($J = 5$). Furthermore we examine three geographic regions ($R = 3$), each region is associated with a clinic location to which it is closest (as shown Figure 3.8). We examine two morbidity types ($M = 2$).

To keep the number of dimensions tractable, we use a statistical relationship between morbidity and region to determine the average number of patients in each population i, as shown in Figure 3.6. This allows us to determine $X_{i,j}$ without accounting explicitly for $I \times J$ decision variables.

The discrete-event simulation model uses data from a variety of sources, including statistical measurements from the Group Health Cooperative data set between $2011-2012$. Based on patient records, we specify a model that describes the arrival rate of patients into the system as well as the use of virtual care, and the probability of using scheduling appointments based on measured statistics. Based on these measurements, we populate a sample model considering the panel re-design of a number of primary care panels within a set of geographically distributed clinics.

Using patient information collected from the Group Health Cooperative, we determine that the average panel contains between $2000-3000$ associated patients. For purposes of patient arrivals, we split the patient population into two morbidity categories based on resource utilization band (RUB) with the low morbidity category comprising patients between 0 and 1 (average) RUB and the high-morbidity category containing patients with (average) RUB between 2 and 3. Based on this breakdown of the patient information we have an average inter-arrival time of 0.015 (hours) and 0.006 (hours) from low and high morbidity respectively (see Table C.7 and C.8).

For the probability of using virtual care, we reference measurements of patient interactions with the Group Health Data set from $2011-2012$ showing that the range of average virtual care interactions ranges from 2% and 15% of all interactions with the medical system. We associate

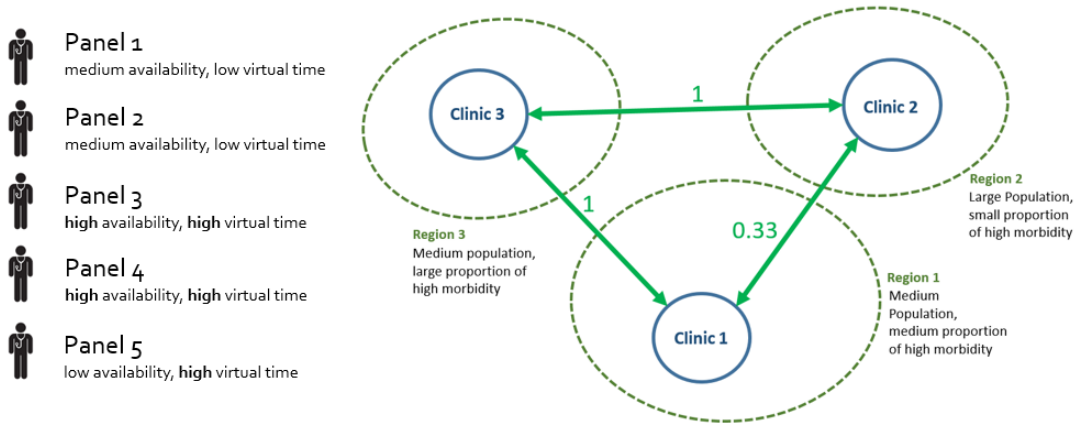

Figure 3.8: Three geographically centered clinics with three associated regions. Each region has a relative distance to a clinic, a different population size, and a different percentage of high and low morbidity patients. There are five physician-panels with different availability, and virtual time characteristics.

the high probability of virtual care with patients with a low general morbidity (0.15) and the lower probability of virtual care with patients that have higher morbidity (0.02).

Similarly, based on literature [46] concerning the probability of no-shows, a probability of a no-show is determined for low and high morbidity patients and is combined with an increase of 20% when a patient is traveling to a distant clinics. This gives us a value of $n_{m=1,r,l} = \{10\%, 12\%\}$ and $n_{m=2,r,l} = \{5\%, 6\%\}$ depending on the distance between region r and clinic l.

Due to the variable use of "open-access" scheduling, we assume that 50% of the patients seek scheduling versus coming to the clinic as "first-come first-serve" patients. Service and schedule time are set based on typical service times. The time frame for each visit appointment is a triangular distribution ($min = 0.4$, $average = 0.5$, and $max = 0.6$ hours). The time for virtual care is set to 15 minutes (0.25 hours). Patients balk after one hour in the queue ($b_i = 1$).

For purposes of the case study, we have the travel weight between a region and clinic nearest to that region to be zero. We associate a base travel weight of 1 between clinics that are far apart and $1/3$ weight for clinics within a closer proximity as shown in Figure 3.8 and Table C.10. The case study could be interpreted as a set of clinics with two downtown locations and one ex-urban clinic location.

Furthermore, we assume that physician availability ranges between 50 and 60 hour work weeks [31]. For each of the five physician panels we set three (1, 2, 5) as "medium availability" (8 hours available a day) and the others (3, 4) to "high availability" (10 daily hours). Furthermore, we set the virtual availability to low (half hour) for two physician panels (1 and 2) and high (2 hours) for three physician panels (3,4,5). This generally corresponds to a pattern of physician work hours that average between 8 and 10 hours a day on average.

Objective function weights are shown in Table C.13. For high morbidity patients we set objective function weights to reflect the greater impact disruptions in care have upon higher morbidity patients. The objective function weights for low-morbidity patients are set to 1. Both virtual wait-time objective weights are set to 1 for both patient populations. On scheduling a conflict the patients jump their scheduled time forward by a random amount (uniformly) between 10 to 16 hours and then attempt to get care again. We use a general time-frame of $t_{frameLimit} = 30$ days as a standard time-frame for measuring the objective function output.

Simulation Optimization

Due to the fact that the policy selection requires choosing both continuous and integer variables, common simulation optimization techniques such as ranking and selection used for discrete domain of decisions may leave out important policy opportunities. We explore several black-box optimization techniques in order to develop a better policy solution for the composition and location of primary care panels. We examine **three different algorithms**:

- (I) Particle Swarm Optimization [PSO] (described in Appendix A)
- (II) Modified Probabilistic Branch and Bound [mod-PBnB] (described in Chapter 5 Section 5.3)
- (III) Nested Partitions with PBnB [NP-PBnB] (described in Chapter 5 Section 5.4)

Particle Swarm Optimization, PSO, is a standard optimization algorithm that performs a random search based on a series of particles that move with random velocity based on the best sampled point [25]. The second algorithm, the "modified probabilistic branch and bound", mod-PBnB, attempts

to locate a specific optimal solution by successively running the PBnB algorithm with decreasing δ in order to focus in on a promising region for the global optimum. Lastly, we explore the "Nested Partition Probabilistic Branch and Bound", NP-PBnB, in order to focus on promising solutions while reducing the required number of partitions that need to be tracked at a given iteration. The full statement of the basic PBnB algorithm is included in Appendix A.3 and the modified versions of the NP-PBnB and mod-PBnB algorithms are discussed in detail in Chapter 5.

We also use a baseline solution that represents the "best-guess". The best guess solution has an equal distribution of the populations between panels, and the panels are assigned equally between the clinics (with only one panel located at clinic 1 due to the odd number of panels). We use this best guess as a useful initial point for the PSO algorithm and for NP-PBnB (mod-PBnB does not use an initial point).

For the PSO algorithm (I), we parameterize the algorithm with $\phi_p = 2$, $\phi_g = 2$, and $\omega = 0.7$ with 10 particles, in order to control how fast the particles move towards local and global optimal points. For the mod-PBnB algorithm we use $\alpha = 0.90$, with $B = 7$, $\varepsilon = 0.10$, we use a starting $\delta_1 = 0.90$ and decrease the value of the δ to $\delta_2 = 0.30$. For the NP-PBnB Partition algorithm, we run the first $k = 1, \ldots, 20$ iterations with minimal sampling with $N_k = B = 7$ and $N_{backtrack} = 10$. We then start an iteration at $k = 21$ with $N_k = B \cdot \frac{ln(\alpha_k)}{ln\left(1 - \frac{\varepsilon_k}{v(\sigma_i)}\right)}$ as outlined in Chapter 5. The selection allows the algorithm to focus in quickly on an initially promising region (on each dimension) and only visiting other regions when a backtrack step is performed. Each of the optimizers generate about 1000 function evaluations. However, due to the way each optimizer iterates, the total number of function evaluations varies slightly. Generally neither algorithms that employ the PBnB algorithm are able to prune regions (exclude them from consideration with a fixed probability) within the allocated computational budget.

The general runtime of the simulation is 20 seconds for one evaluations since each optimizer uses 1000 function evaluations, this requires a runtime of approximately 6 hours for each of the tested algorithms with no additional replications. To deal with the long runtimes, we perform the optimization on a simulation with a *single* replication as in [59] (which can be effective in some circumstances as shown in Chapter 6).

Comparing the optimizer performance in Figure 3.9, we graph the optimizer performance in terms of the best sampled objective value at each function evaluation. We see from the progression

of values that the PSO algorithm almost immediately makes its improvement and then does not shift away from its best point sampled at around the 50th function evaluation. NP-PBnB is able to start at a promising initial point and then make a fast and continuous improvements at later iterations. Starting at a randomly selected value, mod-PBnB makes a fast improvement to an improved solution early in the iterations and makes very few improvement at later iterations.

To compare the final recommendations generated by each of the solutions, we run each of the final solutions for 100 replications to better account for noise. For comparison, we graph the box plots of estimated objective function values in Figure 3.10 with median values listed for each of the three solutions found by each optimizer and our initial "best-guess" solution. The mod-PBnB and the NP-PBnB optimizers generate better solutions, on average, than the "best-guess". The mean objective function value of the best-guess is 8.538 (with 95% confidence interval $[8.494, 8.583]$), the mod-PBnB algorithm and the NP-PBnB algorithm generate policies that are significantly better than the "best-guess" with means 8.134 and 8.291 and with 95% confidence bounds of $[8.110, 8.160]$ and $[8.251, 8.333]$, respectively. However, the PSO algorithm generates a solution that provides a worse policy proposal than the "best guess" (mean 9.099 and 95% confidence interval $[9.025, 9.174]$). This suggests that the PSO optimizer, as run, locates a false local optimum early and does not move from that solution. Overall, the solution generated by mod-PBnB provides the best objective value among available solutions. The average objective function value found by mod-PBnB and NP-PBnB are close, suggesting that there a number of solutions with a similar performance.

Optimization Policy Recommendations and Analysis

Obtaining the final solutions from each of the optimizers, we tabulate the results for each of the decision variables in Tables 3.1, 3.2, 3.3, and 3.4. We also provide a bar chart for the values of $X_{i,j}$ in Figure 3.11, each bar shows the allocation of low morbidity patient populations between each of the panels ($X_{i=1,j}$) in solid, and the allocation of high morbidity patient population between the panels ($X_{i=2,j}$) in stripes. The number in the bar indicates the clinic location for the physician panel. A further examination of the policy solutions for panel placement geographically is shown in Figure 3.12 which shows the assignment of physician-panels to clinics.

Examining graphs and numbers provided for the solutions generated by each of the optimizers, provide a good idea of the recommendations generated by each of the optimizers. First, examining

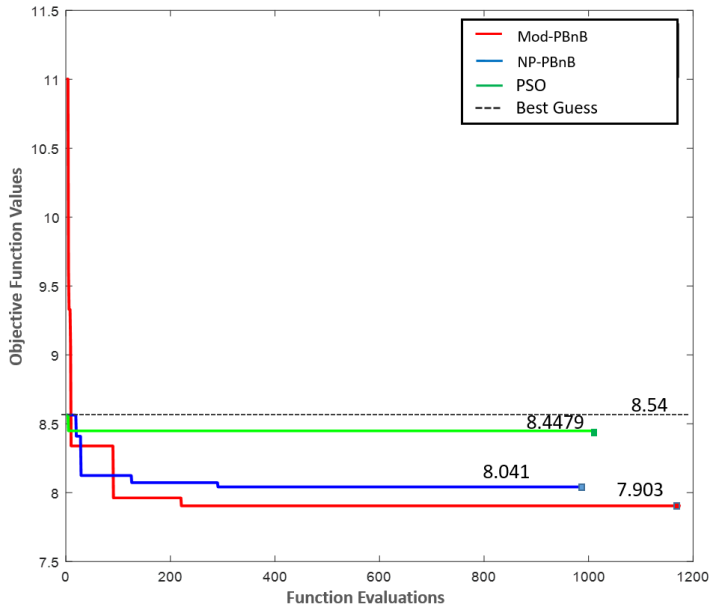

Figure 3.9: The figures shows the progress of the optimization methods used (mod-PBnB, NP-PBnB, and PSO) with a single observation of the objective function per point. The broken line graphs the average objective function value of the "best-guess" over 100 replications. Both PBnB algorithms have similar descent profiles; lower than PSO which levels out early.

the allocation of virtual care, we see generally that the optimized solutions from mod-PBnB and NP-PBnB algorithm give higher hours (V_j) to the panels that can accommodate more hours (indicating that it is generally better to allocate more time for virtual care),.

Examining Figure 3.12, the various policy solutions locate a different number of patient panels at each of the clinics. In the case of clinic 2, we see that the policy recommended by both of the PBnB algorithms, along with the "best-guess" solution, locate two panels there to accommodate the region with the largest population. However, for clinics 1 and 3 the recommendations change with the "best-guess" and the NP-PBnB solutions placing more panels at clinic 3 to accommodate a smaller population with a higher morbidity rate, therefore leaving only a single panel located at clinic 1 to serve a medium sized population with a lower morbidity. In contrast the solution found by mod-PBnB locates two panels at clinic 1 with the medium sized population with a lower morbidity.

The difference in panel allocation is likely due to the fact that the NP-PBnB focuses its sampling within nested sets around an initial "best guess", this biases the solution towards allocations similar

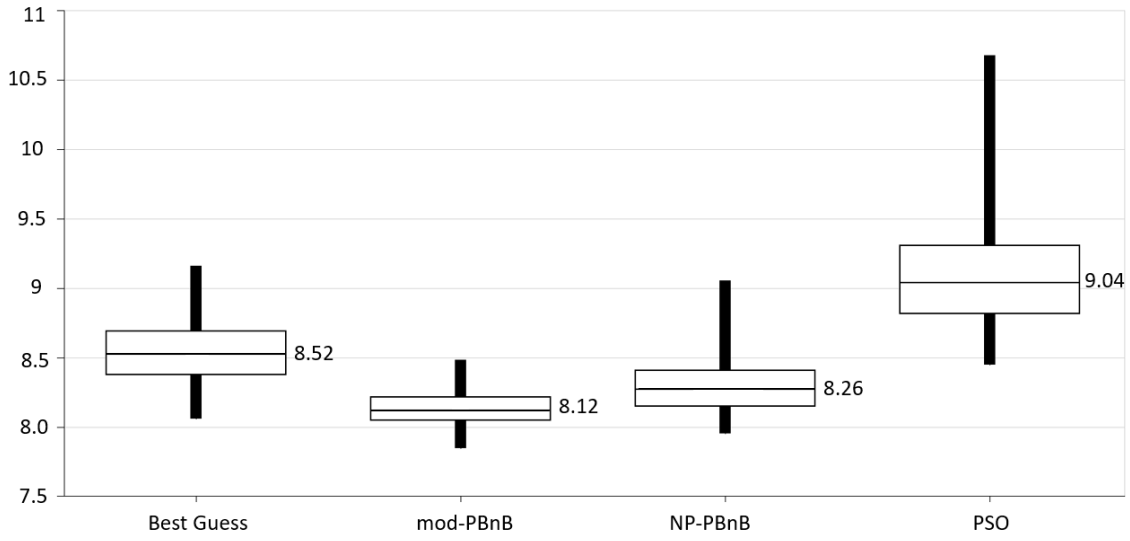

Figure 3.10: Graphed box plots for each of the generated policy solutions: Best-Guess, mod-PBnB, NP-PBnB, and PSO. The medians of 100 replications are labeled with the boxes representing $+/-$ 25% and the bold lines indicate the minimum and maximum values (excluding outliers).

to "best-guess" equal distribution of patient panels in the system. This indicates the advantages of algorithms that spend their computation budget exploring the domain generally rather than searching in the proximity of an initial point.

The distribution of the patient population among the different panels also differ as shown in Figure 3.11. The solutions provided by the mod-PBnB algorithm and the NP-PBnB algorithm shown somewhat similar distributions of patient populations with the higher capacity panels (3 and 4) having a greater proportion of the patient populations. However, some key factors in the policy recommendations persist. Here, for the mod-PBnB the high capacity panel 4 located at clinic 3 takes on a large percentage of the high-morbidity patients (reducing travel time due to region 3 having a large percentage of high morbidity patients).

The mod-PBnB solution locates the other large capacity panel 4 at clinic 3 to serve the large volume of low morbidity patients. By contrast the solution provided by the NP-PBnB algorithm places two of the lower capacity panels at clinic 3 and allocates a larger proportion of the higher morbidity patients between them. This is essentially a different approach to replacing the load of high morbidity patients, either through handing them largely through a few dedicated panels

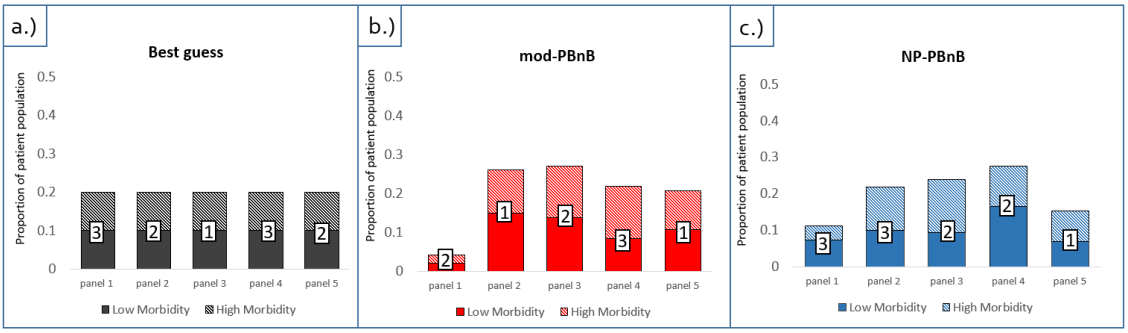

Figure 3.11: The policy solution generated from each of the optimizers (a. best guess, b. mod-PBnB, c. NP-PBnB). The PSO result is not graphed. The bars show the distribution of the low morbidity patients (solid) and the high morbidity patients (striped) between the five physician-panels. The clinics associated with the physician-panels are labeled in white boxes.

Table 3.1: The location (clinics $1,2,3$) of each of the physician panels, Y_j ($j = 1,\ldots,5$). Each row corresponds to the final solution provided by the respective optimization method.

optimization method	**panel 1**	**panel 2**	**panel 3**	**panel 4**	**panel 5**
"best guess"	clinic 3	clinic 2	clinic 1	clinic 3	clinic 2
"modified PBnB"	clinic 2	clinic 1	clinic 2	clinic 3	clinic 1
"NP-PBnB"	clinic 3	clinic 3	clinic 2	clinic 2	clinic 1
"Particle Swarm"	clinic 2	clinic 2	clinic 2	clinic 2	clinic 1

or dividing them up among several smaller capacity panels located in high morbidity areas. The objective function values of these two solutions are close, although the 95% confidence intervals do not overlap.

One unusual development in the solutions of both PBnB algorithms is the low allocations of patient populations to panel 1. In both cases the panel shares a clinic with another panel to which more of the overall patient demand is directed. This may point to a problem with the objective function which weighs lowering the travel time and wait time of the patient and does not place priority on balancing the load between panels overall. Further updates to the objective might consider further weighing deviations from the desired utilization rate on each of the panels or including a further penalty for radically unbalanced patient distribution.

The policy recommendations generally show the importance of properly considering the morbidity of the population under consideration, in addition to considering the importance of virtual

Table 3.2: The proportion of the low-morbidity patients ($i = 1$) assigned to each of the physician panels, $X_{i=1,j}$, ($j = 1,\ldots,5$). Each row corresponds to the final solution provided by the respective optimization method.

optimization method	**panel 1**	**panel 2**	**panel 3**	**panel 4**	**panel 5**
"best guess"	0.2000	0.2000	0.2000	0.2000	0.2000
"modified PBnB"	0.0413	0.2992	0.2778	0.1677	0.2140
"NP-PBnB"	0.1454	0.2002	0.1872	0.3295	0.1378
"Particle Swarm"	0.2525	0.1074	0.1373	0.2430	0.2598

Table 3.3: The proportion of the high-morbidity patients ($i = 2$) assigned to each of the physician panels, $X_{i=2,j}$, ($j = 1,\ldots,5$). Each row corresponds to the final solution provided by the respective optimization method.

optimization method	**panel 1**	**panel 2**	**panel 3**	**panel 4**	**panel 5**
"best guess"	0.2000	0.2000	0.2000	0.2000	0.2000
"modified PBnB"	0.0429	0.2216	0.2640	0.2719	0.1996
"NP-PBnB"	0.0812	0.2364	0.2902	0.2217	0.1705
"Particle Swarm"	0.1271	0.4180	0.1711	0.0199	0.2639

Table 3.4: The number of virtual hours scheduled for each of the physician panels, V_j ($j = 1,\ldots,5$). Each row corresponds to the final solution provided by the respective optimization method.

optimization method	**panel 1**	**panel 2**	**panel 3**	**panel 4**	**panel 5**
"best guess"	0.5000	0.5000	0.5000	0.5000	0.5000
"modified PBnB"	0.4460	0.3369	1.0362	1.0096	1.8862
"NP-PBnB"	0.3023	0.3917	1.2502	1.2013	1.1133
"Particle Swarm"	0.4033	0.4788	0.5831	1.1383	0.5173

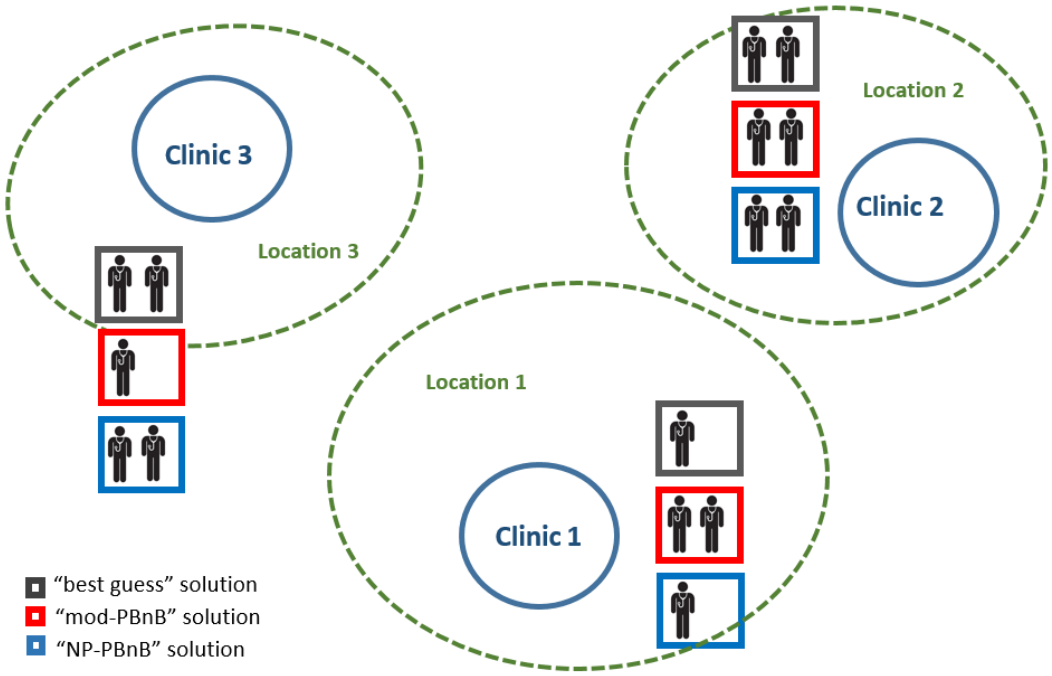

Figure 3.12: The assignment of panels to clinics for three solutions: the "best-guess" (gray), the mod-PBnB (red), and the NP-PBnB (blue).

care use when placing primary care physician-panels. Further models may wish to include a larger variety of different morbidity types, additional panels, and a more diverse set of regions for considerations. However, using the full number of panels and regions in the data set (8 panels per clinics across 7 clinics and 7 regions for a total of 56 physician-panels) would result in a model of upward of 200 variables which would result in a significantly longer runtime $30-60$ seconds for a single observation, and a greater number of function evaluations needed to obtain a result that has a lower objective function value than a "best-guess" policy that equally distributes each patient population between the panels.

This problem points to the general need for optimization algorithms that can develop good solutions (close to global optimum) for high dimensional problems. In Chapter 4, we extend several theoretical algorithms that have provable performance in high dimensions and discuss implementations of these methods in Chapter 5 with normalized variation.

Chapter 4

ADAPTIVELY SAMPLING FROM NESTED LEVEL SETS FOR GLOBAL OPTIMIZATION

As seen in Chapter 3, a critical feature of effective global optimization algorithms is to develop a relatively good solution relatively quickly. This issue becomes compounded for problems with a large number of domain dimensions. Ideally, we would like to have a finite-time analysis of optimizer performance that shows the computation does not increase exponentially as the domain dimension increases. Two central issues are addressed in this chapter. First, an analysis of how the addition of noise impacts the performance of an adaptive random search algorithm as a function of dimension is presented. Second, the analysis is extended to adaptive random search in terms of sampling within measurable level sets by quantiles, as commonly used by partition-based algorithms.

In this chapter, consider an optimization problem,

$$\min_{x \in S} f(x) \tag{P0}$$

where S is a closed and bounded subset of $\mathbb{R}^n$, $x \in S \subset \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$. Denote the minimum value and optimal point on the domain as:

$$y_* = \min_{x \in S} f(x) \text{ and } x_* = \arg\min_{x \in S} f(x). \tag{4.1}$$

Denote the maximum value and point as:

$$y^* = \max_{x \in S} f(x) \text{ and } x^* = \arg\max_{x \in S} f(x). \tag{4.2}$$

We allow for multiple optimal (minimum) solutions. Furthermore, for a domain S, we define a diameter d as the diameter of S, or the greatest distance between any two points in S.

While finite-time performance analyses are rare for most commonly used optimization algo-

rithms, finite-time analyses are available for several theoretical adaptive random search algorithms which demonstrate that, under certain conditions, the number of iterations required to sample below a specified objective function value increases only linearly in dimension when optimizing a function without noise [17, 87, 92, 103, 105, 107, 108].

To quantify finite-time performance as a function of domain dimension, Pure Adaptive Search (PAS) (see Appendix A) can be analyzed. The PAS algorithm samples from an improving set of points at each iteration. Finite-time analysis establishes that, under certain conditions, the expected number of iterations until PAS achieves a value close to the minimum increases only linearly in terms of the dimension of the domain [107, 108]. However, the requirement that PAS improves at every iteration makes it difficult to implement practically.

Hesitant Adaptive Search (HAS) generalizes PAS by allowing for "hesitation" where the algorithm does not improve with a certain specified probability that depends on the best objective function value sampled [106, 108]. HAS is defined with a sampling distribution ζ on S and a bettering probability $b(y)$, defined for $y_* \leq y \leq y^*$. The probability of hesitation is $1-b(y)$. Therefore, HAS can be defined as follows:

Hesitant Adaptive Search (HAS), cf. [17]

- **Step 0:** Initialize X_0 in S according to a sampling distribution ζ. Set $k=0$. Set $\bar{Y}_0 = f(X_0)$.
- **Step 1:** Generate X_{k+1} from the normalized restriction of ζ on the improving set $S_{k+1} = \{x \in S : f(x) < \bar{Y}_k\}$ with probability $b(\bar{Y}_k)$, and set $\bar{Y}_{k+1} = f(X_{k+1})$. Otherwise, set $X_{k+1} = X_k$ and $\bar{Y}_{k+1} = \bar{Y}_k$.
- **Step 2:** If a stopping criterion is met, stop. Otherwise, increment k and return to Step 1.

Finite-time analysis of HAS for a closed form expression for the expected number of iterations until reaching a specified $\varepsilon > 0$ above the minimum function value [17, 103] as,

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{d\rho(t)}{b(t) \cdot p(t)}. \tag{4.3}$$

where $\rho(y) = \zeta(f^{-1}([-\infty, y]))$, and $p(y) = \rho((-\infty, t])$. PAS is a special case of HAS, when the bettering probability equals one, and sampling distribution ζ is uniform.

This chapter describes an extension of Hesitant Adaptive Search to a problem where the objective function must be estimated and provides finite-time results in Section 4.1 Theorem 3 provides an expression for the expected number of iterations required to obtain a desired value ε above the minimum (analogous to (4.3)). Corollary 4 shows that under certain conditions the expected number of iterations is bounded by a linear function of the domain dimension. Theorem 5 bounds the number of function evaluations (including replications) to obtain a value that is ε above the minimum by a cubic function of the domain dimension. We consider this chapter to be a completion of research objective 2(a).

In Section 4.2, we introduce a new adaptive random search, called Quantile Adaptive Search, that focuses on sampling points within a set of nested quantile level sets. The general motivation of examining quantiles is to extend the finite-time analysis to algorithms that sample from a set of nested quantile level sets. This provides insight into computational potential for algorithms that focus on sampling from level sets with quantile estimators.

4.1 Hesitant Adaptive Search with Estimation (HAS-E)

In this section, we consider an optimization problem where the objective function, $f(x)$, cannot be observed directly but must be estimated. Let $f(x) = E[g(x,\chi)]$ where $g(x,\chi)$ is a function on $x \in S$ and χ, a random variable. The function $g(x,\chi)$ may be evaluated in a discrete event simulation. The optimization problem is:

$$\min_{x\in S} f(x) = \min_{x\in S} \; E[g(x,\chi)] \tag{P1}$$

where S is a closed and bounded subset of $\mathbb{R}^n$ and $x \in S \subset \mathbb{R}^n$. We denote the minimum and maximum of (P1) as in (4.1) and (4.2).

Given that the value of $f(x)$, for $x \in S$, cannot be directly observed, the value has to be estimated by performing a certain number of replications. To estimate the function value, we consider $g(x,\chi_r)$ for a number of replications $r = 1,\ldots,R$ at a point x. The sample mean estimate (dropping the x for convenience) is:

$$\hat{y}^{est} = \frac{\sum_{r=1}^{R} g(x,\chi_r)}{R}. \tag{4.4}$$

We assume that $\hat{y}^{est} \sim N(f(x), \frac{\sigma}{\sqrt{R}})$, where $\sigma^2 = Var(g(x,\chi))$. This will hold under the central limit theorem for large R. A confidence interval on $\hat{y}^{est}$ can be developed with an upper bound $\hat{y}^{high}$ with confidence $1-\alpha$, $0 \leq \alpha \leq 1$, where:

$$\hat{y}^{high} = \hat{y}^{est} + \frac{\sigma \cdot z_{\alpha/2}}{\sqrt{R}}. \tag{4.5}$$

Where $z_{\alpha/2}$ is the standard normal value at $\alpha/2$. We note that $\hat{y}^{high} \sim N(f(x) + \frac{\sigma \cdot z_{\alpha/2}}{\sqrt{R}}, \frac{\sigma}{\sqrt{R}})$ and therefore

$$P\left(f(x) \leq \hat{y}^{high} \leq f(x) + \frac{\sigma \cdot z_{\alpha/2}}{\sqrt{R}}\right) \geq 1-\alpha.$$

which provides an $(1-\alpha)$ confidence bound for the location of $\hat{y}^{high}$.

Since the function of the sampled point cannot be directly observed, the sampling of additional points needs to be based on the estimated values of the function with replications. The HAS-E algorithm uses the estimate $\hat{y}^{high}$ to focus sampling on regions that are likely to be improving. This algorithm is called Hesitant Adaptive Search with Estimation (HAS-E), and is fully specified as follows.

On the kth iteration of HAS-E, the sampled point $x_k \in S$ is used to define $y_k = f(x_k)$ as the true value, the estimate as $\hat{y_k}^{est}$ (with R_k replications as defined in (4.4)), and upper bounds $\hat{y_k}^{high}$ as in (4.5). Subsequently, we denote the level set S_y is given as:

$$S_y = \{x \in S : f(x) < y\} \tag{4.6}$$

We also denote, the respective level sets of these values S_{y_k}, $S_{\hat{y_k}^{est}}$, and $S_{\hat{y_k}^{high}}$ as in (4.6). The general approach of HAS-E is to track the best sampled true value at iteration k, $\bar{y}_k$, and sample from the level set defined by the estimated upper-bound, i.e., $S_{\bar{y}_k^{high}}$, as shown in Figure 4.1.

Hesitant Adaptive Search with Estimation (HAS-E)

Set input parameters α, σ, γ along with a sampling distribution ζ with support on the entire domain S.

- **Step 0:** Sample X_0 in S according to the probability distribution ζ on S. Set $k = 0$ conduct R_0

replications of the function at the initial selected point (i.e., $g(x,\chi_r)$ for $r = 1,\dots R_0$), estimate the value $\hat{y}_0^{high}$ as in (4.5) and set $\bar{y}_0^{high} = \hat{y}_0^{high}$.

- **Step 1:** Sample X_{k+1} according to the normalized restriction of ζ on the set $S_{\bar{y}_k^{high}}$ with "bettering" probability γ, otherwise set $X_{k+1} = X_k$. Perform R_k replications at X_{k+1} (if $X_{k+1} \neq X_k$) and estimate $\hat{y}_{k+1}^{high}$ as in (4.5). Then update the value of $\bar{y}_{k+1}^{high}$, such that

$$\bar{y}_{k+1}^{high} = \hat{y}_{k'}^{high}$$

where $k' = argmin_i \{f(x_i) : i = 0,\dots,k+1\}$. Then set,

$$\bar{Y}_{k+1} = f(X_{k'}) \tag{4.7}$$

to update the algorithm with the best value encountered.

- **Step 2:** If a stopping criterion is met, stop. Otherwise, increment k and return to Step 1.

As with HAS, the HAS-E algorithms is a framework for analysis, and not practical to implement. However, the framework allows us to provide finite time analysis for the algorithm's performance. The analysis beings with Theorem 1, describing the number of replications chosen on each iteration. We next prove in Theorem 2 that under certain assumptions about the replications, HAS-E stochastically dominates a special case of HAS without estimation. Theorem 3 provides a bound for the expected number of iterations and Corollary 4 provides a bound on the expected number of iterations increases linearly in dimension, under certain conditions. Finally, a bound on the expected number of replications needed to obtain this result is given in Corollary 5.

For purposes of our analysis, the ratio of volumes $\frac{\nu(S_{y_k})}{\nu(S_{\hat{y}_k^{high}})}$ requires a lower bound, where $\nu(\cdot)$ is the n-dimensional volume of a set. A trivial bound for this quantity can be based on the value $\varepsilon > 0$ where $y_k > y_* + \varepsilon$, such that

$$\frac{\nu(S_{y_k})}{\nu(S_{\hat{y}_k^{high}})} \geq \frac{\nu(S_{y_*+\varepsilon})}{\nu(S)}$$

which holds regardless of the number of replications that are used to make the estimate.

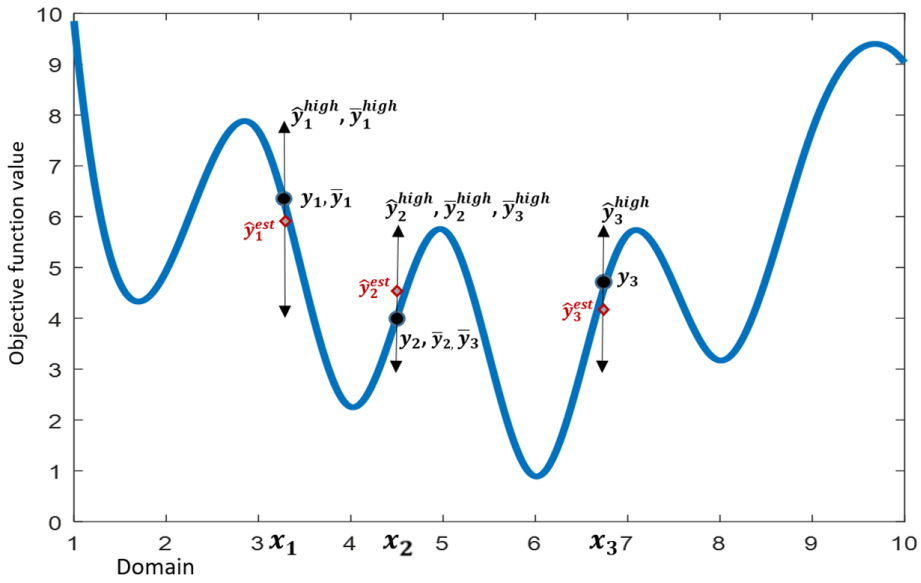

Figure 4.1: An illustration of sampling across three iterations ($k = 1,2,3$). The sampled points are labeled x_1, x_2, and x_3, with values y_1, y_2, and y_3. The estimated values for the upper bound $\hat{y}_1^{high}, \hat{y}_2^{high}$, and $\hat{y}_3^{high}$ are also shown. The values $\bar{y}_1, \bar{y}_2$, and $\bar{y}_3$ correspond to the best sampled value with corresponding upper-bounds $\bar{y}_1^{high}, \bar{y}_2^{high}$, and $\bar{y}_3^{high}$.

However, we develop a further bound based on the number of replications such that $\frac{v(S_{y_k})}{v(S_{\hat{y}_k^{high}})} \geq q$ for any value $0 < q < 1$ provided that a large enough number of replications R are used in the estimation of $\hat{y}_k^{high}$.

The general concept is illustrated in Figure 4.1 where the ratio of the volumes between the level set S_{y_k} and $S_{\hat{y}_k^{high}}$ becomes closer to 1 as the distance between the values $\hat{y}_k^{high}$ and y_k decrease; as the number of replications increase, the distance between $\hat{y}_k^{high}$ and y_k decreases until $\frac{v(S_{y_k})}{v(S_{\hat{y}_k^{high}})}$ can be bounded below for a selected value q. Theorem 1 provides a bound on the number of replications to achieve a lower bound q.

For a function $f(x)$ on domain S, for a given $0 < q < 1$, let κ_q be the maximum value such that, for any z, $y_* < z < y^*$, and any $\kappa_q' < \kappa_q$, then $\frac{v(S_z)}{v(S_{z+\kappa_q'})} > q$. Based on this quantity and the diameter d of the domain S, we define

$$\mathscr{K}_q = \frac{\kappa_q}{d}.$$

The quantity $\mathscr{K}_q$ can be viewed as the ratio of the change in objective function to the diameter

of S that is associated with q. Furthermore, we define $\mathscr{B}_y$ as the largest ball, centered at x_* that can be inscribed inside a level set S_y for $y_* < y < y^*$ with radius r_y. Using these two defined concepts, we can relate the number of replications to a selected q.

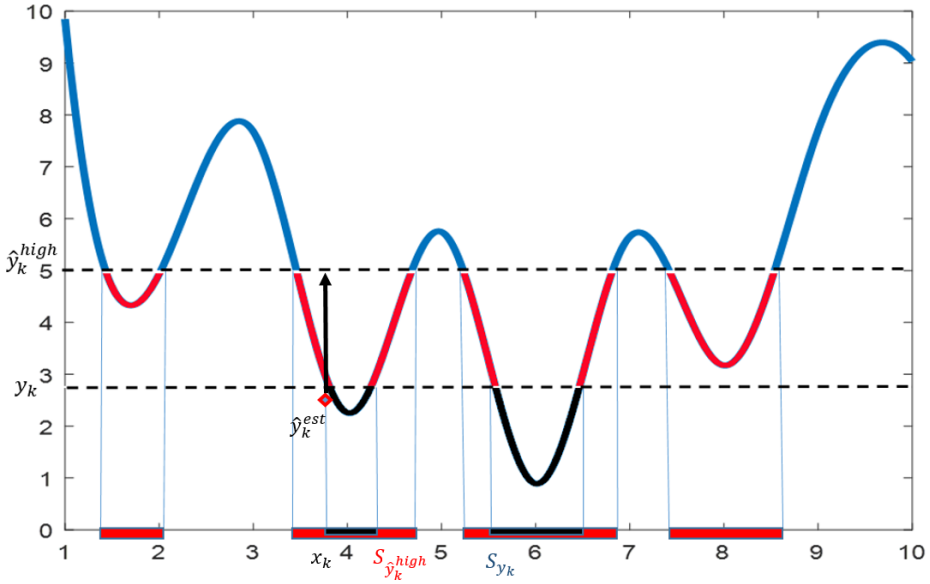

Figure 4.2: An illustration of the values y_k and $\hat{y}_k^{high}$ along with their corresponding level sets S_{y_k} and $S_{\hat{y}_k^{high}}$ for a one-dimensional problem. The level sets are shown highlighted on the horizontal axis. The ratio between the volumes of the level sets increases as the difference between $\hat{y}_k^{high}$ and y_k decreases (which happens with a large number of replications).

Theorem 1. *Given a function, f, and a point x_k in the domain with a value $y_k = f(x_k)$, and $\hat{y}_k^{high}$ estimated with $R_k = R$ replications, for any given value $0 < q < 1$ and $\varepsilon > 0$ such that $y_k \geq y_* + \varepsilon$ if*

$$R \geq \left(\frac{\sqrt[n]{q} \cdot 2 \cdot z_{\alpha/2} \cdot \sigma}{(1 - \sqrt[n]{q}) \cdot r_{y_* + \varepsilon} \cdot \mathscr{K}_q} \right)^2 \tag{4.8}$$

then:

$$\frac{\nu(S_{y_k})}{\nu(S_{\hat{y}_k^{high}})} \geq q \tag{4.9}$$

when $y_k \leq \hat{y}_k^{high} \leq y_k + \frac{2 \cdot \sigma \cdot z_{\alpha/2}}{\sqrt{R}}$ (which occurs with probability $(1 - \alpha)$).

Proof: See the proof outlined in Section B.2 of Appendix B.

We go on demonstrate that the HAS-E stochastically dominates a special case of HAS and go on to provide a bound for the expected number of iterations and replications. Theorem 2 states that HAS-E stochastically dominates a special case of the standard HAS algorithm.

We assume that the sampling distribution ζ has an lower and upper bound on the density over S denoted ζ_{low} and ζ_{high}. This implies that ζ has a non-zero probability of sampling anywhere on the domain. The special case, HAS1, samples uniformly on the improving level set, $S_{\bar{y}_k}$, where $S_{\bar{y}_k} = \{x \in S : f(x) < \bar{y}_k\}$ and $\bar{y}_k = min\,(f(X_0), \ldots, f(X_k))$. The HAS1 bettering probability is chosen to be $(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma$. Based on this definition of HAS1, Theorem 2 proves stochastic dominance of HAS-E over HAS1.

Theorem 2. *Consider problem (P1). Let $\bar{Y}_k^{HASE}$ be the best sampled value on the kth iteration of the HAS-E algorithm with sampling distribution ζ and constant bettering probability γ. Let $\bar{Y}_k^{HAS1}$ be the best sampled value of HAS1, the special case of the HAS algorithm with bettering probability $(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma$ and uniform sampling distribution. $\bar{Y}_k^{HASE}$ stochastically dominates $\bar{Y}_k^{HAS1}$, that is:*

$$P(\bar{Y}_k^{HASE} \leq y) \geq P(\bar{Y}_k^{HAS1} \leq y) \text{ for } k = 0, 1, \ldots$$

where $y_* < y \leq y^*$.

Proof: The proof is provided in Section B.2 of Appendix B.

Using Theorem 2 we go on to prove that the HAS-E algorithm has a well-defined upper bound on the expected number of iterations until it samples within a set level of the true minimum y_* through a direct comparison to the HAS-E algorithm. Based on the dominance of the HAS-E algorithm the number of iterations to obtain a desired objective function value necessarily have to be fewer than the corresponding basic HAS algorithm with constant bettering probability and uniform sampling.

Theorem 3. *Given HAS-E then an upper bound on the expected number of iterations until reaching a value of $y_* + \varepsilon$ is:*

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{d\rho(t)}{(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma \cdot p(t)}$$

$$= 1 + \frac{1}{(1-\alpha)\cdot\left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma} \cdot ln\left(\frac{\nu(S)}{\nu(S_{y_*+\varepsilon})}\right) \tag{4.10}$$

Proof: See the proof outlined in Section B.2 of Appendix B.

Corollary 4. *When S in (P1) is a convex feasible region in n dimensions with a diameter d and $f(x)$ satisfies the Lipschitz condition with Lipschitz constant at most $\mathscr{K}$, then the expected number of iterations for HAS-E to reach a value $y_* + \varepsilon$, $\varepsilon > 0$, is bounded by,*

$$E(N(y_* + \varepsilon)) \leq 1 + \left(\frac{n}{(1-\alpha)\cdot\left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma}\right) \cdot ln\left(\frac{\mathscr{K}\cdot d}{\varepsilon}\right). \tag{4.11}$$

Proof: The expression in (B.9) combined with the bounds on $\left(\frac{\nu(S)}{\nu(S_{y_*+\varepsilon})}\right)$ in [103, 106] produce the result. □

Corollary 5. *For HAS-E with $R_k = R$ replications for each iteration,*

$$R = \left(\frac{\sqrt[n]{q}\cdot 2\cdot z_{\alpha/2}\cdot\sigma}{(1-\sqrt[n]{q})\cdot r_{y_*+\varepsilon}\cdot\mathscr{K}_q}\right)^2 \tag{4.12}$$

the expected number of function evaluations to achieve a value within ε of the minimum $\varepsilon > 0$, $E(R(y_* + \varepsilon))$ is upper-bounded by a cubic function of domain dimension:

$$\begin{aligned} &E\left(R(y_* + \varepsilon)\right) \\ &\leq \left(1 + \left(\frac{n}{(1-\alpha)\cdot\left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma}\right) \cdot ln\left(\frac{\mathscr{K}\cdot d}{\varepsilon}\right)\right) \cdot \left(\left(\frac{q}{1-q} + \frac{-log(q)}{(1-q)^2}\cdot n\right)\frac{2\cdot z_{\alpha/2}\cdot\sigma}{r_{y_*+\varepsilon}\cdot\mathscr{K}_q}\right)^2 \\ &\sim O\left(n^3\right). \end{aligned}$$

Proof of Corollary 5

Since, the number of replications each iteration is constant, we write the expected number of replications as,

$$E\left(R_k(y_* + \varepsilon)\right) = R \cdot E(N(y_* + \varepsilon)).$$

Applying the inequality in Lemma 12 from Appendix B with Theorem 1, we get:

$$\left(\left(\frac{q}{1-q}+\frac{-log(q)}{(1-q)^2}\cdot n\right)\frac{2\cdot z_{\alpha/2}\cdot\sigma}{r_{y_*+\varepsilon}\cdot\mathscr{K}_q}\right)^2 > R = \left(\frac{2\cdot\sqrt[n]{q}\cdot z_{\alpha/2}\cdot\sigma}{(1-\sqrt[n]{q})\cdot r_{y_*+\varepsilon}\cdot\mathscr{K}_q}\right)^2.$$

Combining this with (4.11) from Corollary 4 for $E(N(y_*+\varepsilon))$ we obtain

$$E(N(y_*+\varepsilon))$$

$$\leq\left(1+\left(\frac{n}{(1-\alpha)\cdot\left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2\cdot q\cdot\gamma}\right)\cdot ln\left(\frac{\mathscr{K}\cdot d}{\varepsilon}\right)\right)\cdot\left(\left(\frac{q}{1-q}+\frac{-log(q)}{(1-q)^2}\cdot n\right)\frac{2\cdot z_{\alpha/2}\cdot\sigma}{r_{y_*+\varepsilon}\cdot\mathscr{K}_q}\right)^2$$

which is a cubic equation with respect to the dimension n. □

This corollary demonstrates that the number of replications needed to obtain a value of ε above a given value is bounded by a cubic *polynomial* function of the dimension. This results generally extends the good finite-time results of the HAS framework for problems with estimation. In the next section, we go on to examine a framework based on an adaptive search framework that samples from a series of nested quantile level sets.

4.2 The Quantile Adaptive Search (QAS)

In this section, we define a new algorithm called Quantile Adaptive Search (QAS) which is an optimization theoretical framework that attempts to locate optimal solutions for the problem *P0*, without noise, by sampling from a series of nested level sets associated with decreasing quantiles. The algorithm utilizes a series of sampling distributions defined by density function as ζ_k for $k = 0,\ldots,K$ (on iteration k) such that $\zeta_k(x) > 0$ for all $x \in S$.

We are interested in sampling solutions in a series of level sets defined by quantiles. For a quantile level, $0 < \delta < 1$, let a level set be denoted:

$$L(\delta,S) = \{x : f(x) \leq y(\delta,S)\}$$

where $y(\delta, S)$ is the δ-quantile of the domain S, or explicitly:

$$y(\delta, S) = \underset{y_* < y < y^*}{\arg\min} P(f(X) \leq y) \geq \delta$$

where X is a random variable uniformly distributed on S. We assume that the sampling densities, ζ_k, are non-negative and bounded, with $\zeta_k^{high} = max(\zeta_k(x) : x \in S)$ and $\zeta_k^{low} = min(\zeta_k(x) : x \in S)$ for $k = 1, \ldots, K$. The probability of sampling a point X_k from the distribution ζ_k within the δ quantile level set, $L(\delta, S)$, is denoted:

$$P(X_k \in L(\delta, S)) = \int_{L(\delta,S)} \zeta_k(x)dx = P(Y_k \leq y(\delta, S))$$

where $Y_k = f(X_k)$. Based on these definitions, we outline the QAS algorithm.

Quantile Adaptive Search (QAS)

Step 0: Start with parameters $\mathscr{C} > 1$ and $0 < \gamma \leq 1$. Initialize $k = 0$ with δ_0 with a given sampling density ζ_0. Sample X_0 from ζ_0. $\bar{Y}_0 = f(X_0)$

Step 1: Sample X_{k+1} from the sampling distribution ζ_{k+1}, where ζ_{k+1} and δ_{k+1} satisfy four conditions:

(i) $\delta_{k+1} \leq \mathscr{C} \cdot \delta_{(k)}$, where $\delta_{(k)}$ is the quantile defined by record value $\bar{Y}_k$

(ii) $P(\{X_{k+1} \in L(\delta_{k+1}, S)\}) \geq \gamma$

(iii) $P(\{X_{k+1} \in L(\delta, S)\} | \{X_{k+1} \in L(\delta_{k+1}, S)\}) \geq P(\{X_0 \in L(\delta, S)\} | \{X_0 \in L(\delta_{k+1}, S)\})$

(iv) $P(\bar{Y}_{k+1} < y | \bar{Y}_k = \bar{y}_k)$ is non-increasing in $\bar{y}_k$ for $y < \bar{y}_k$

Set

$$\bar{Y}_{k+1} = \begin{cases} f(X_{k+1}), & \text{if } f(X_{k+1}) < \bar{y}_k \\ \bar{Y}_k, & \text{otherwise} \end{cases}$$

Step 3: If a stopping criterion is met, stop. Otherwise, increment k and return to Step 1.

The motivating approach used for the QAS is that the random search will successively improve its probability of sampling within a set of decreasing quantile level sets at every iteration. The criteria for selecting ζ_{k+1} and δ_{k+1} as specified in (i) - (iv) constrains the target quantile and sampling distribution with the following requirements:

(i) *The quantile δ_{k+1} is proportional to the quantile $\delta_{(k)}$ associated with upper bound of the estimate the best observed value $\bar{y_k}$.*

(ii) *The conditional probability of the kth sampling distribution, ζ_k of sampling within the δ_k quantile level set $L(\delta_k, S)$ is bounded below by some minimum probability γ.*

(iii) *The conditional probability that the distribution ζ_k samples within lower level sets given it samples within $L(\delta_k, S)$ is not lower than that of ζ_0.*

(iv) *The conditioned probability that the distribution ζ_k samples within a lower level set given that the previous sampled value was y_k is non-decreasing in y_k.*

The challenge with implementation is selecting the δ-quantile values for which a sampling distribution can sample within the respective quantile level set. By constructing the algorithm in terms of a decreasing quantile δ_k and an associated sampling distribution ζ_k, the QAS controls the rate of sampling with the quantile parameter δ_k and distribution ζ_k, just as the Annealing Adaptive Search is able to obtain the same finite time results as HAS by controlling the temperature parameter T_k.

To illustrate the general form of *QAS* see Figure 4.2. The three level sets at decreasing quantile levels $\delta_{k+2} < \delta_{k+1} < \delta_k$ so that $L(\delta_{k+2}, S) \subset L(\delta_{k+1}, S) \subset L(\delta_k, S)$. At each iteration, a new "target" quantile δ_k is selected along with a sampling distribution ζ_k. The sampling distribution ζ_k is selected to keep some minimum probability of sampling within the "target" level set $L(\delta_k, S)$. Therefore, the selection of a target quantile δ_k can be seen as mechanism for focusing the sampling distribution on targeted quantile level sets.

We now provide an upper bound on the expected number on iterations until QAS samples a value within $\varepsilon > 0$ of the optimal value y_*, $E(N(y_* + \varepsilon))$. A general approach is to prove that the general case of *QAS* stochastically dominates a special case of *HAS*.

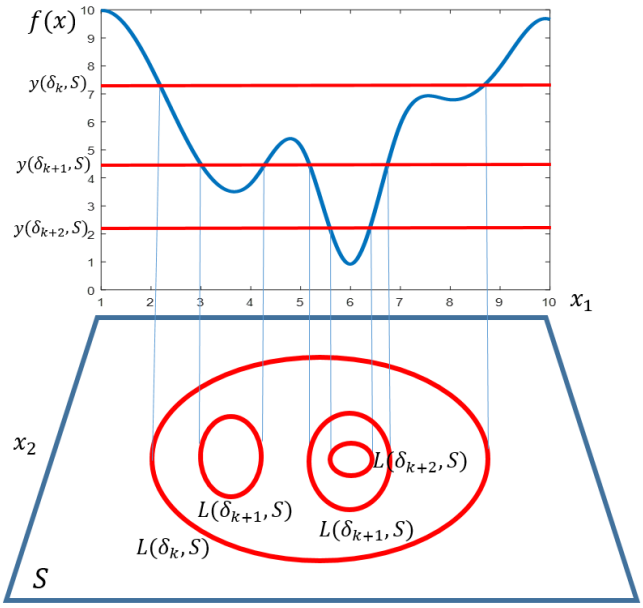

Figure 4.3: An illustration of series of nested nested level sets with $\delta_k > \delta_{k+1} > \delta_{k+2}$. QAS seeks to sample from a set of level sets on each iteration.

With a QAS algorithm for some probability γ and constant $\mathscr{C}$ and an upper and lower bound on the initial sampling probability, ζ_0, such that $\zeta_0^{low} \leq \zeta_0(x) \leq \zeta_0^{high}\ \forall x \in S$. Consider the objective function values $\bar{Y}_k^{QAS}$ as a stochastic process on iteration k. Denote $\bar{Y}_k^{QAS}$ as the best sampled value from *QAS*.

Given *QAS*, we specify a special case of the *HAS* algorithm, called HAS2, with a constant bettering probability b equal to $\left(\frac{\gamma}{\mathscr{C}}\right) \cdot \left(\frac{\zeta_0^{low}}{\zeta_0^{high}}\right)^2$ and sampling distribution ζ_0 restricted to the improving level set, and best sampled value $\bar{Y}_k^{HAS2}$. Based on these definitions, we show that *QAS* stochastically dominates *HAS*2.

Theorem 6. *Consider the problem (P0). Let $\bar{Y}_k^{QAS}$ be the best sampled point at the kth iteration with parameters γ and $\mathscr{C}$. Let $\bar{Y}_k^{HAS2}$ be the best sampled of HAS2, the special case of HAS with bettering probability, $b = \frac{\gamma}{\mathscr{C}} \cdot \left(\frac{\zeta_0^{low}}{\zeta_0^{high}}\right)^2$ and a sampling distribution ζ_0. $\bar{Y}_k^{QAS}$ stochastically dominates $\bar{Y}_k^{HAS2}$:*

$$P(\bar{Y}_k^{QAS} \leq y) \geq P(\bar{Y}_k^{HAS2} \leq y) \text{ for } k = 0, 1, 2, \ldots,$$

where $y_* < y < y^*$.

Proof: See the proof outlined in Section B.3 of Appendix B.

Theorem 7. *The expected number of iterations until the value of* $y_* + \varepsilon$ *or better is sampled by the QAS algorithm can be given an upper bound of:*

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{\mathscr{C}}{\gamma} \cdot \left(\frac{\zeta_0^{high}}{\zeta_0^{low}} \right)^2 \cdot \left(\frac{d\rho(t)}{p(t)} \right) \tag{4.13}$$

Proof: See the proof outlined in Section B.3 of Appendix B.

Corollary 8. *If the initial sampling distribution* ζ_0 *is uniform on the domain and* f *is a Lipschitz continuous function with constant* $\mathscr{K}$ *the expected number of iterations until the value of* $y_* + \varepsilon$ *or better is sampled by the QAS algorithm,*

$$E(N(y_* + \varepsilon)) \leq 1 + n \cdot \frac{\mathscr{C}}{\gamma} \cdot \left(\frac{\zeta_0^{high}}{\zeta_0^{low}} \right)^2 \cdot ln\left(\frac{\mathscr{K} \cdot d}{\varepsilon} \right) \tag{4.14}$$

where d *is the diameters of the domain* S.

Proof: The expression in (4.13) combined with the bounds on $\left(\frac{v(S)}{v(S_{y_*+\varepsilon})} \right)$ in [103, 106] produce the result $\square$

This proves that, for the QAS algorithm, the expected number of iterations that are needed to reach a specified value above the global minimum increases linearly in dimension. Although this is a very strong result for the QAS framework, there are many difficulties in practically implementing the QAS framework for a realistic problem. One issue is determining the value of the quantile level associated with the best sampled point $\delta_{(k)}$, the next is determining an algorithm that can consistently sample within specified quantiles.

In the next chapter, we explore algorithms that can sample within a specified quantiles by partitioning the domain. These algorithms, under certain circumstances, attempt to approximate the ideal performance of Quantile Adaptive Search on high dimensional problems.

Chapter 5

ADAPTIVE SEARCH IMPLEMENTATION USING PARTITION BASED ALGORITHMS ON HIGH-DIMENSIONS

Implementing the QAS algorithm has many difficulties similar to implementing Pure Adaptive Search or Hesitant Adaptive Search. First, there is the problem of determining a sampling distribution that can sample within a specified quantile δ_k, second there is the difficulty approximating the quantile level that defines the best sampled value $\delta_{(k)}$. However, QAS provides motivation for developing algorithms that can sample from various level sets within a domain. If the algorithm is able to sample from a quantile level set consistently at a target quantile, δ_k, an aggressive lowering of the level δ_k on each iteration k may be able to approximate the ideal performance of the QAS framework. Under these conditions, the developed implementations should have good performance for higher dimensional problems.

There are a number of partition-based algorithms that can be used to attempt to sample within a δ quantiles. Using these algorithms, paired with some method of lowering the δ_k targeted by the algorithms at each iteration, this approach will allow optimization algorithms to sample widely across the domain on early iterations, and then focus the random search into more promising regions with a higher density of near-optimal points. These methods include the Nested Partitions framework which focuses sampling on sub-regions of a domain based on some "promising" index [96], Probabilistic Branch and Bound (PBnB) which statistically approximates a level set [105, 104], and Optimal Computational Budgeting applied to partition-based optimization [21]. By exploring partition-based algorithms that follow this general form, we may be able to address problems of sampling within a specific level set.

In addition to describing some other novel new to partition-based optimization, the section goes on to use PBnB for the basis of two new algorithms. PBnB is able to approximate quantile level-sets with a series of hyper-rectangles (algorithm specified in Section A.3). Previous research on the PBnB algorithm [104] provides finite time results that statistically guarantee certain regions

are inside a target δ quantile level set, called "maintained", or outside, called "pruned" within a volume margin of error ε at some confidence level $(1-\alpha)^4$. This result allows algorithms that use Probabilistic Branch and Bound to determine, in real time, whether the algorithm has sampled within a level set, which is a key insight or users. This research uses a shifting targeted quantile, with the hopes that the modified PBnB algorithm can be used to approximately implement QAS. We consider this chapter to be a completion of research objective 2(b).

This chapter presents four implementable partition-based algorithms that attempt to improve the performance of global optimization on high dimensional problems. Section 5.1 details an extension of Optimal Computational Budgeting Allocation (OCBA) algorithm for continuous random search, incorporating a "Look-ahead" methodology to improve the performance on multi-dimensional problems, this is based on research in [54].

In section 5.2, based on [55], we detail an extension of the Nested Partitions that focuses on the sampling points within a targeted quantile, where we are able to extend several of the theoretical convergence properties of the regular Nested Partition algorithm to our algorithm.

In Section 5.3 we present a straight-forward application of the PBnB algorithm with a shifting quantile level to further focus sampling within promising regions of the domain, modified Probabilistic Branch and Bound (mod-PBnB). In Section 5.4, we combine a Nested Partition framework with PBnB to describe an algorithm called Nested Partition Probabilistic Branch and Bound (NP-PBnB).

5.1 Optimal Computational Budget Allocation with Lookahead

Broadly there is a large range of OCBA methods to provide efficient use of function evaluations by algorithms. Research done by Chen et al. [20, 21] is widely used in ordinal optimization [18, 19]. Only recently have OCBA algorithms been discussed in terms of partition-based optimization methodology [54].

We have extended previous work done on OCBA in order to prioritize taking samples across a multi-dimensional problem for a partition-based random search. Under our modified method, a lookahead approximation is used in order to estimate the amount of budget that should be allocated to optimize each dimension. The algorithm uses the approximation of probability of correctly se-

lecting a partition in one dimension to determine the order of dimensional-search and a stopping criterion for each dimension in the partition-based random search. The key insight in this contribution is to use a "remaining probability of correct selection" to estimate the relative utility of performing one dimension before the other, or allocating more of the total budget to developing a certain approximation of the currently optimized dimension over saving that budget for later dimensions. This subsection is selected from previous research published in the Proceedings of the 2015 Winter Simulation conference [54].

We start with the problem

$$min_{x\in\Theta} f(x) \tag{5.1}$$

where f is function defined on a domain Θ with dimensions D. The algorithm partitions each dimension sequentially and then select the best region within that partition.

We can express a set of subregions of the space Θ such that each dimension is divided into M equal intervals so that the space is partitioned into M^D boxes $\theta_\gamma^{(D)}$ indexed by a D length array of tuples $\gamma = ((d_1, m_{d_1}), (d_2, m_{d_2}), ..., (d_D, m_{d_D}))$ where $d_1, ..., d_D$ is a permutation of the dimensions $\{1, ..., D\}$, and the values $m_{d_1}, ..., m_{d_D}$ range from 1 to M and indicate a specific interval on a dimension. We define a box of order k as

$$\theta^{(k)}_{(d_1,m_{d_1}),...,(d_k,m_{d_k})} = \bigcup_{m_d=1}^{M} \theta^{(k+1)}_{(d_1,m_{d_1}),...,(d_k,m_{d_k})(d,m_d)}$$

where d is one value in the set of remaining dimensions $\mathbb{D}_r^{(k)} = \{1, ..., D\}/\{d_1, d_2, ..., d_k\}$ and $k = 0, 1, ..., D-1$. For reference, $\theta^{(k)}_{(d_1,m_{d_1}),...,(d_{k-1},m_{d_{k-1}})(d_k,\cdot)}$ indicates the set of boxes $\{\theta^{(k)}_{(d_1,m_{d_1}),...,(d_{k-1},m_{d_{k-1}})(d_k,m)}$ $\forall m = 1, ..., M\}$.

The approach determines the best order of dimensions to optimize as shown in Figure 5.1 for a three dimensional example. A partition-based algorithm will have to select which dimension to divide and select first, here the example shows the algorithm dividing along dimension 2 first (gray), followed by dimension 3 (dark gray), followed by dimension 1 (black).

For a set of correct indices ${d_1}^*, ..., m^*_{d_D}$ the Probability of Correct Selection (PCS) is denoted $P(m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_D} = m^*_D)$ for any permutation $d_1, ..., d_D$ of $\{1, ..., D\}$. The Optimal Computational

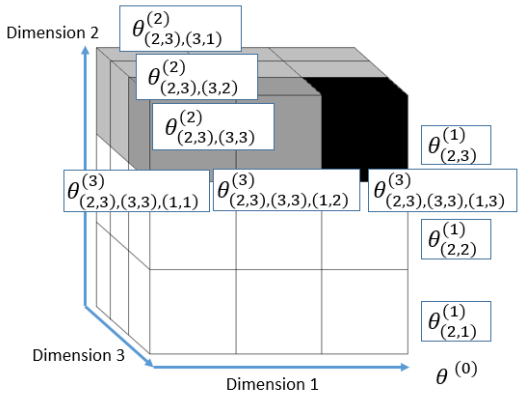

Figure 5.1: An example of boxes of different order in a three dimensional domain, with $M = 3$. The order to the dimensional division will result in a different set of sub-regions and a different selection problem. The look ahead determines the effectiveness of ordering the selection of dimensions.

Budget Allocation (OCBA) problem can be defined as follows. Let $N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}$ be the number of samples in box $\theta^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}$. Our goal is to select the allocation of samples to maximize the probability of correct selection relative to some budget T. The problem then becomes a question of maximizing the probability of correct selection by selecting the number of samples and ordering of dimensions.

$$\begin{aligned} &max_{N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}} PCS|N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})} \\ &\text{subject to} \quad \sum_{m_{d_1}=1}^{M} \cdots \sum_{m_{d_D}=1}^{M} N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})} = T, \text{ for any permutation } d_1,\ldots,d_D \text{ of } \{1,\ldots,D\} \end{aligned} \tag{5.2}$$

One solution to reducing the size of problem (5.2) is to sequentially optimize the probabilities of correct interval selection on each dimension. In sequence, each probability can be conditioned on selecting previous dimensions' interval correctly. The result is a series of D optimization problems across M variables corresponding to each dimension where the kth iteration determines the optimal $m^b_{d_k}$ which is used to condition the next $k+1$st order problem.

We can describe this function as the probability of correct selection based on allocating budget

across dimensions in a certain order. To accomplish this, our research proposes a heuristic called *Approximate Probability of Correct Selection*, that depends only on the choices the budget allocated to already optimized dimensions

$$APCS(k, d_k, T_{d_k}, ..., T_{d_D}). \tag{5.3}$$

In order to approximate a lower bound, we assume that all future budget is equally distributed across the unspecified dimensions and that the probability of correctly selecting a dimension is based on the first k divisions of the domain. A further lower bound is created by estimating $P(m_d^b = m_d^*)$ using observed samples in the $k+1$ order boxes where $d_{k+1} = d$. Therefore at every iteration, we compute $APCS_{leave}$ and $APCS_{stay}$, the approximated probability of correct selection based on whether we continue to allocate budget to selecting the optimal interval in this dimension or reserving the remaining budget for optimizing along the remaining dimensions. Using lookahead heuristics, the algorithm is better able to divide up a computational budget in between dimensions.

The initial methods for using a lookahead approximation, has generated some benefits to the computational efficiency of partition-based methods. For a problem that is not symmetrical in dimension, the lookahead provides some benefits to developing an efficient solution. This general result illustrates a significant advantage to employing the modified OCBA algorithm to the problem of optimizing a multidimensional black box function. Furthermore, the algorithm might be easily applied to our goals in research objective *2(a)*. However more work is needed to apply similar OCBA methodology to the risk-based optimization model.

5.2 Quantile Based Nested Partition Algorithm

Although, OCBA is effective at improving the efficiency of random search algorithms, it does not directly address the question of sampling from a quantile in a way that might approximately implement *QAS*. However, a number of partition-based algorithms can be easily extended to sample from quantile level sets. We first examine an extension of the Nested Partition algorithm that targets quantiles [95]. We include the following section from a paper published in the Proceedings of the 2016 Winter Simulation conference [55].

In this section we describe the implementation of a quantile-based Nested Partition algorithm for black-box functions. We prove that the quantile-based Nested Partition algorithm converges in probability to an unbranchable region (e.g., specified by the user) with the smallest value of the targeted quantile. We also prove that, with a sufficiently low target quantile, the probability that the unbranchable region with the lowest estimated quantile contains the true global optimum approaches 1 as the number of iterations become arbitrarily large. We implement the quantile-based Nested Partition algorithm on a set of common test problems with and without OCBA sampling as in [21]. We find that the addition of the OCBA sampling scheme results in better (closer to the global minimum) sampled points using the same allocated budget.

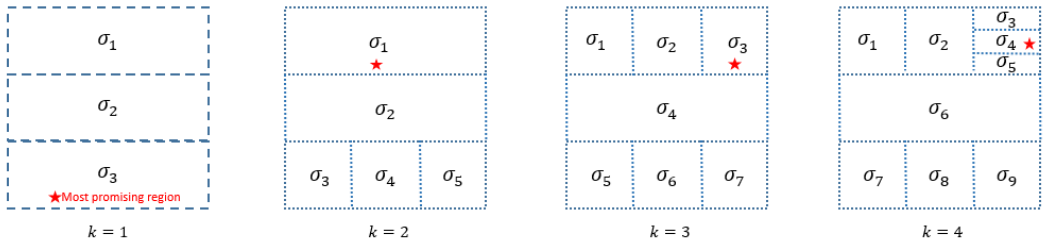

Figure 5.2: An example illustrating four iterations on a two dimensional domain, with $M = 3$. Each iteration the algorithm samples from a lower level-set. Here the selected "most-promising" region is marked with a red star.

Given a partitioning scheme (typically along each dimension), we define a region as "unbranchable" when the length of each dimension i is less than or equal to some length ε_i. Other definitions for unbranchable might include a minimum volume or a contained diagonal length. Let Σ_{max} denote the set of unbranchable regions that form a partition on S (i.e., $\bigcup_{\sigma \in \Sigma_{max}} \sigma = S$ and $\sigma \bigcap \sigma' = \emptyset$).

The inputs to the algorithm include the δ-threshold quantile, the branching scheme M, a defined budget per iteration T, and a minimum number of iterations K. The input parameter K sets a minimum number of iterations that is typically larger than the number of iterations to achieve an unbranchable set by consecutive partitioning, to prevent the algorithm from stopping prematurely. A maximum number of iterations may also be set, however we use the discovery of an unbranchable region to determine the stopping condition. The full algorithm is written below:

Quantile-based Nested Partition Algorithm (QNP)

STEP 0 Initialize: Set δ, M, T, and K. Set $\Sigma_{contend}(0) = \{S\}$. Define the most promising region (best) $\sigma^B(0) = S$ and set $k = 1$.

STEP 1 Partition: If $\sigma^B(k-1)$ is unbranchable then $\Sigma_{contend}(k) = \Sigma_{contend}(k-1)$. Otherwise, partition the most promising region, $\sigma^B(k-1)$, into M regions of equal volume $\sigma^B(k-1)_1, \ldots, \sigma^B(k-1)_M$, and update

$$\Sigma_{contend}(k) = \left(\Sigma_{contend}(k-1) \backslash \sigma^B(k-1)\right) \bigcup_{m=1}^{M} \sigma^B(k-1)_m.$$

Let $\sigma_j^k, j = 1, \ldots, \|\Sigma_{contend}(k)\|$ represent each region in $\Sigma_{contend}(k)$.

STEP 2 Sample: Sample N_j^k points from region in $\sigma_j^k \in \Sigma_{contend}(k)$ for $j = 1, \ldots, \|\Sigma_{contend}(k)\|$ including previously sampled points such that $\sum_{j=1}^{\|\Sigma_{contend}(k)\|} N_j^k = k \cdot T$. Note there must be at least one newly sampled point in each of the regions in $\Sigma_{contend}(k)$ (we specify a method for setting the budgeting allotments N_j^k using OCBA or otherwise). Denote the sampled points in σ_j^k as

$$x_1^j, \ldots, x_{N_j^k}^j.$$

Rank the sample points by their function evaluations, i.e., $x_{(1)}^j, \ldots, x_{(N_j^k)}^j$ such that $f(x_{(1)}^j) \leq f(x_{(2)}^j) \leq \cdots \leq f(x_{(N_j^k)}^j)$.

STEP 3 Estimate Quantile: Let $\mathbf{v}_{min}(k)$ be the smallest volume of the regions in $\Sigma_{contend}(k)$. For each $\sigma_j^k \in \Sigma_{contend}(k)$ for $j = 1, \ldots, \|\Sigma_{contend}(k)\|$ determine $\hat{y}_j$ as an estimate of the quantile $y\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right)$, estimated as:

$$\hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right) = f\left(x^j_{\left(ceil\left(N_j^k \cdot \delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}\right)\right)}\right)$$

where the expression $ceil(x)$ is the lowest integer greater than x.

STEP 4 Rank: Determine a new most promising region $\sigma^B(k) \in \Sigma_{contend}(k)$ such that $\hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma^B(k))}, \sigma^B(k)\right) < \hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right)\ \forall \sigma_j^k \in \Sigma_{contend}(k)$. In the case of a tie, such that $\hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_i^k)}, \sigma_i^k\right) = \hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right)$ for sets $\sigma_i, \sigma_j \in \Sigma_{contend}(k)$ then let $\sigma^B(k)$ be the set with the greater volume, if volumes are tied break the tie arbitrarily.

STEP 5 Stopping Condition: Record the minimum incumbent value $f_k^B = min_j f(x_{(1)}^j)$ for $j = 1, \ldots, \|\Sigma_{contend}(k)\|$. If $k \geq K$ and $\sigma^B(k)$ is unbranchable then stop the algorithm, otherwise increment k and go to Step 1.

The algorithm proceeds at each iteration to partition the most-promising region if it is unbranchable. It then samples a positive number of points (determined by a budgeting scheme) and estimates quantiles for each of the remaining regions. For purposes of selecting the most promising region, larger regions have proportionally smaller quantiles estimated (relative to the volume of the smallest contending region) and therefore the algorithm has a probability of selecting larger volumes as the "most promising" and "backtracking" to other areas of the domain for consideration. The algorithm terminates once an unbranchable region has been developed and a minimum number of iterations have elapsed. Therefore the algorithm ends with an unbranchable region with the lowest estimated quantile.

An example is shown in Figure 5.2 for a two dimensional domain. On the first iteration the algorithm creates $M = 3$ different regions and selects the most promising region indicated by a $\star$. On the second iteration the most promising region is partitioned (replacing σ_3 with σ_3, σ_4 and σ_5). In this example, the most promising region on iteration 2 is σ_1, illustrating backtracking. The algorithm finally lands on the most promising region σ_4 at $k = 4$ with σ_4 being an unbranchable region.

The quantile-based Nested Partition algorithm has a number of useful convergence properties that mirror the original Nested Partition algorithm. First, we can observe that, as the number of iterations approaches infinity, the most promising region will be the subregion in Σ_{max} with the lowest true quantile. Let σ^* be the best unbranchable region, $\sigma^* \in \Sigma_{max}$ such that $y(\delta, \sigma^*) < y(\delta, \sigma)$ for all $\sigma \in \Sigma_{max}$ with $\sigma \neq \sigma^*$. For purposes of analysis we assume σ^* is unique.

Theorem 9. *As $k \to \infty$ then $P\left(\sigma^B(k) = \sigma^*\right) \to 1$.*

Proof: See the proof outlined in Section B.4 of Appendix B.

Theorem 10. *If f is a function that satisfies the Lipschitz condition with Lipschitz constant L, there exists a value δ^* such that for all $\delta < \delta^*$ then $P(x^* \in \sigma^B(k)) \to 1$ as $k \to \infty$.*

Proof: See the proof outlined in Section B.4 of Appendix B.

Although the quantile-based Nested Partition algorithm is guaranteed to eventually find the unbranchable region with lowest specified quantile, the efficiency of the algorithm will largely depend on the allocated budget N_j^k in each contending region $\sigma_j^k \in \Sigma_{contending}(k)$. At each iteration k the algorithm will sample points and branch the most promising region in order to focus more sampling in the newly branched regions on the next iteration. It is therefore important for efficiency to choose the most promising region with the lowest quantile to minimize backtracking and focus sampling in the regions likely to contain optimal points.

To ensure efficiency of the algorithm, values for N_j^k are chosen to maximize the probability of correctly selecting the most promising region. At a given iteration, we define an index b such that $y\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right) \geq y\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_b^k)}, \sigma_b^k\right) \quad \forall \sigma_j^k \in \Sigma_{contend}^k$ for $j \neq b$.

Because of the normality of the estimator, we can extend the methodology for OCBA across a partitioned domain in order to asymptotically maximize the probability of correctly selecting the most promising region. The formula derived by [19] specifies equations that asymptotically optimize the probability of correct selection when choosing sampling points to divide between a discrete number of normally distributed estimators,

$$\frac{N_{j'}}{N_j} = \left(\frac{\frac{s_j}{\delta_{b,j}}}{\frac{s_{j'}}{\delta_{b,j'}}}\right)^2 \quad \forall j, j' \neq b \qquad N_b = s_b \sqrt{\sum_{j=1, j\neq b}^{\|\Sigma_{contend}(k)\|} \frac{N_j^2}{s_j^2}}$$

QNP allows for the sampling of points within a level-set and can be used in combination with an recursive algorithm to implement QAS with the right selection of δ at each iteration. However, the quantile based Nested Partitions framework, does not have results that extend to changing quantile

levels. The next sections explore extending the convergence results of this algorithm when the δ quantile level is changed from iteration to iteration.

5.3 Modified Probabilistic Branch-and-Bound

The Probabilistic Branch and Bound (PBnB) algorithm has a finite time analysis and relatively good performance for level-set approximation of a black box function on a continuous domain with noise [45]. The algorithm generates a set of samples inside a set of hyper-rectangles and prunes and maintains regions based on the statistical likelihood that each of the regions will fall within the specified level-set. Through a number of iterations, the algorithm guarantees that a minimum volume ε is incorrectly pruned from the level set with probability $(1-\alpha)^4$. Each iteration involves estimating the upper and lower bounds of the targeted quantile $\hat{y}^{up}(\delta_k, \sigma)$ and $\hat{y}^{low}(\delta_k, \sigma)$. The full PBnB algorithm is listed in Appendix A.

The Probabilistic Branch and Bound algorithm can be used for purposes of optimization by isolating a promising level-set and sampling points within that region of the domain. However, there are direct trade-offs concerning the targeted level quantile δ. If the δ value is too low, the algorithm will not be able to successfully isolate a region for sampling, if δ is too high, the algorithm will not be able to improve sampling within a region that is close to the optimal solution.

Since PBnB constructs a quantile level set approximation, it is a good algorithm for attempting to sampling within a *series* of quantile level sets. To implement we propose modifying the algorithm with a deceasing value of δ_k between iterations. Regions pruned with a set confidence $(1-\alpha)^4$ that they do not contain more than ε of the targeted level set, will also be pruned at the lower level-set δ_{k+1} since $\delta_{k+1} \leq \delta_k$ and $L(\delta_{k+1}, S) \subset L(\delta_k, S)$.

The algorithm can be seen as an extension of the PBnB algorithm with level set approximation, with an additional step outlined in Figure 5.4. Here the a single iteration of the PBnB algorithm is contained in STEP 2 and 3 with additional steps providing rules for updating the target quantile δ_k. The general framework updates the δ_k such that the upper and lower bound of quantile value are estimable. The algorithm requires preset values $\delta_1 \ldots \delta_L$ and $K_1 \ldots K_L$ that control the amount of sampling spent on each quantile level. The new algorithm is defined as follows:

Modified PBnB for Nested Quantile Level Sets (mod-PBnB)

STEP 0 Initialize: Set $k = 1$, $l = 1$, select an initial δ_0, and α, ε, k_b, B, R^0 for the Probabilistic Branch and Bound, and set quantile levels, $\delta_1 \dots \delta_L$. Set sub-iteration limit $K_1 \dots K_L$, $K'_1 \dots K'_L$, and $N_1 \dots N_k$ and max iterations K_{max}. Set $\Sigma_0^{contend} = S$ and $\Sigma_0^{maintain} = \emptyset$ and $\Sigma_0^{prune} = S$. Sample initial set of points uniformly on the domain S.

STEP 1 Determine Contending: Set $l = l + 1$. If the quantile bounds, $\hat{y}^{up}(\delta_k, \sigma)$ and $\hat{y}^{low}(\delta_k, \sigma)$ are estimable, update delta $\delta_k = \delta_l$ and set

$$\Sigma_k^{contend} = \Sigma_{k-1}^{contend} \bigcup \Sigma_{k-1}^{maintain}$$

$$\Sigma_0^{maintain} = \emptyset$$

STEP 2 Sample: Sample a number of points N_k on $\Sigma_k^{contend}$, then run Probabilistic Branch and Bound for one iteration at level δ_k to update $\Sigma_k^{contend}$, Σ_k^{prune}, and Σ_k^{prune} (as outlined in Appendix A.3).

STEP 3 Stopping Conditions: Set $k = k + 1$. If there are no pruned regions within K_l iterations or K'_l iterations have elapsed, return to Step 1, otherwise return to Step 2. If no branchable sub-regions or if $\frac{\varepsilon}{v(S)} < \delta_k$, END the algorithm and report the result.

Initial numerical experiments, have shown some good results for using mod-PBnB as an optimization method. We show initial results on the two dimensional Rosenbrock function in Figure 5.5. Using the values of $\delta_1 = 90\%$, $\delta_2 = 30\%$, and $\delta_3 = 5\%$, we are able to successively sample within decreasing level sets as shown in Figure 5.3.

The mod-PBnB algorithm successfully samples within decreasing quantile level sets as defined by a user. Furthermore, the decreasing level sets allow the algorithm to focus on the promising regions of the domain. The theory attached to the PBnB algorithm generally ensures that we will not incorrectly prune more than ε of the domain at any given iteration.

However, the mod-PBnB algorithm has many of the same problems that the original level-set approximations does. First as the iterations increase, the number of sub-regions needed to track the number of contending regions increases exponentially. Furthermore, mod-PBnB spends a large

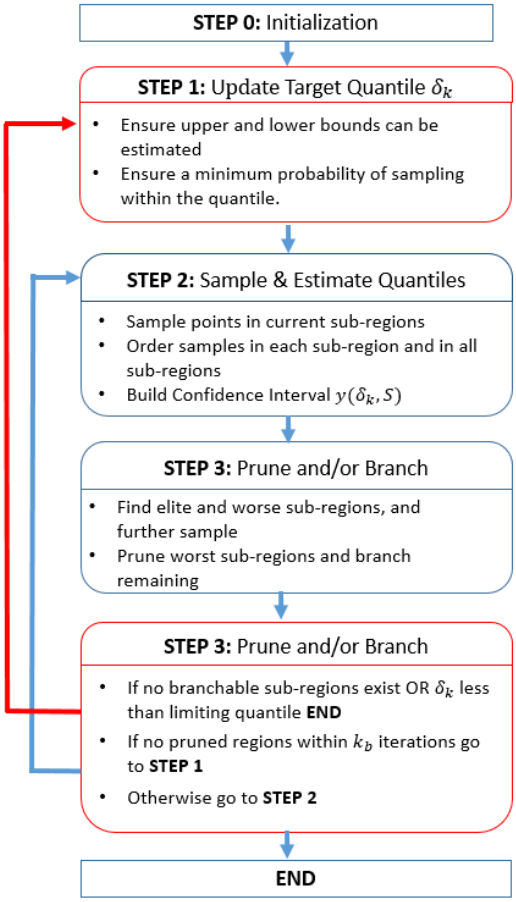

Figure 5.3: An outline of using the mod-PBnB algorithm with a lowering δ_k level at each iteration. The modified portions of the PBnB algorithm are outlined in red which controls the lowering of δ_k.

amount of function evaluations exploring very high quantiles. The mod-PBnB algorithm cannot make use of initial "good" points to focus its sampling efforts. To confront some of these issues, we develop an alternative algorithm that builds the PBnB pruning and maintaining steps into a nested framework in Section 5.4. Here the algorithm limits the total number of rectangles needed to focus sampling on regions that contain a good initial point.

5.4 Probabilistic Branch and Bound in the Nested Partition Framework

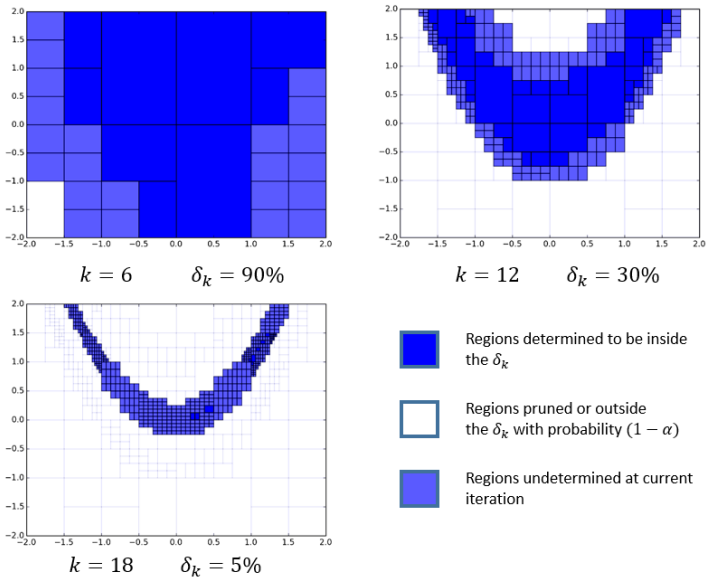

Figure 5.4: An example illustrating three iterations of the mod-PBnB algorithm on a two dimensional Rosenbrock function, with $M = 3$ with maintained regions are in deep blue, contending regions in blue, and pruned regions in white.

Incorporating the learnings from both Sections 5.2 and 5.3, we present one more implementable algorithm that builds the exploration of points into the Nested Partition framework. The general approach, will incorporate the mechanisms for pruning and maintaining regions incorporated into the Nested Partition framework mechanism for focusing on specific regions. This algorithm maintains a fixed number of sub-regions in a partition-scheme and updates itself by trying to estimate the "most-promising" region (based on a promising index function) based on whether the region is likely to be part of the target quantile level-set, and sub-dividing the domain within that new promising region, back-tracking when necessary.

We define a promising index function, such that, for some final quantile δ_k, regions that are inside the level set $L(\delta_k, S)$ are ranked first (using the best value sampled in each region as a tie breaker) followed by the regions that cannot be pruned or maintained in $L(\delta_k, S)$ (using the best value sampled in each region as a tie breaker). This is very similar to selecting regions based on their best point sampled, since as the number of sub-regions goes to infinity a sufficiently small δ_k will result in the most promising region necessarily containing the global optimum (for Lipschitz

continuous functions).

Each iteration, the algorithm then divides the region into B sub-regions. The algorithm then selects a most promising region based on what is pruned and maintained. After selecting, the algorithm will restart the partitioning and selection process on new most-promising region, attempting to sample within the δ quantile of the new region. At each iteration, the algorithm will check to see if the current most-promising region can be pruned from the level set approximation of $L(\delta_K, S)$, ensuring that it constantly attempts to sample from a lower quantile level-set. Every iteration will lower the $\delta_{k+1} = \delta_{k+1}/B$ focusing the sampling down on lower level sets.

In addition to the general combination, the algorithm also performs a look-ahead (as in Section 5.1) by splitting the region on each each dimension and then selecting the dimension for partitioning based on how many pruned or maintained regions will be generated. This look ahead helps the algorithm deal with domains that have asymmetrical properties between different dimensions.

Probabilistic Branch and Bound in Nested Partition (NP-PBnB)

STEP 0 Initialize: Set parameters B divisions, ε, α and target quantile value δ as well an iteration limit K_{limit}. We set $\Sigma_0^C = \{S\}$ and $\Sigma_0^P = \Sigma_0^M = \emptyset$. Set $k = 1$ (iterations) and set the modeled, effective quantile $\delta_1 = \delta$. Set $\Sigma_0^C = \emptyset$ the current contending regions, the identified promising regions $\Sigma_0^M = \emptyset$, and the pruned regions, $\Sigma_0^P = \emptyset$.

STEP 1 Sample: Sample N_k points on each region $\sigma \in \Sigma_{k-1}^C$ (by construction this is one region) and order all of the points on the region, $x_{i,(1)}, x_{i,(2)}, \ldots, x_{i,(N_k)}$, such that:

$$f(x_{i,(1)}) \leq f(x_{i,(2)}) \leq \ldots, f(x_{i,(N_k)})$$

Find the indexes,

$$r_{down} = max_r : \sum_{i=0}^{r-1} \binom{N_k}{i} (\delta)^i \cdot (1-\delta)^{(N_k - i)} \leq \frac{\alpha_k}{2} \text{ and } s_{up} = min_s : \sum_{i=0}^{s-1} \binom{N_k}{i} (\delta)^i \cdot (1-\delta)^{(N_k - i)} \geq \frac{\alpha_k}{2}$$

and estimate quantile thresholds, so that $\hat{y}^{up}(\delta, \sigma) = f(x_{i,(s_{up})})$ and that $\hat{y}^{low}(\delta, \sigma) = f(x_{i,(r_{down})})$.

STEP 1 Backtrack: Sample $N_{backtrack}$ additional points on S, order all sampled points $\{x\}$ with or-

dered points $\{x'_{(1)} \dots x'_{(k \cdot N_{backtrack})}\}$ and corresponding values $\{y'_{(1)} \dots y'_{(k \cdot N_{backtrack})}\}$, estimate the upper and determine the upper and lowerbound for the region S, where:

$$s'_{up} = min_s : \sum_{i=0}^{s-1} \binom{N_k}{i} (\delta_k)^i \cdot (1-\delta_k)^{(N_k - i)} \geq \frac{\alpha_k}{2}$$

and that $\hat{y'}^{up}(\delta_k, \sigma) = y'_{(s'_{up})}$.

- If $y^n_{i,(1)} > \hat{y}^{up'}(\delta_k, \sigma)$, $k > 1$, and $\sigma \neq S$. Then set $\Sigma^P_k = \Sigma^P_{k-1} \bigcup \Sigma^C_{k-1}$ and $\Sigma^C_k = \sigma_i$ where σ_i is the largest volume region in Σ^C_{k-1} and $\sigma_i \notin \Sigma^P_k$. Set $k = k+1$, and $\delta_k = M \cdot \delta_{k-1}$ then go to **STEP 1**.

- Else proceed to **STEP 2**.

STEP 2 Branch: For each dimension, n, create a partition of the region, $\sigma \in \Sigma^C_k$ of B subregions, such that $\sigma = \{\sigma^n_1 \bigcup \sigma^n_2 \dots \bigcup \sigma^n_B\}$. For each region, index the N^n_k sampled values within it, such that

$$\{y^n_{i,(1)} \dots y^n_{i,(N^n_k)}\} \in \sigma^n_i$$

Define, $N_{\sigma^n_i}$ as the number of points sampled in σ^n_i. If $N_{\sigma^n_i} > \frac{ln(\alpha_k)}{ln\left(1 - \frac{\varepsilon_k}{v(\sigma)}\right)}$, determine whether each of the regions is pruned or promising such that

$$M^n_i = \begin{cases} 1, & \text{if } y^n_{i,(N^n_k)} < \hat{y}^{low}(\delta, \sigma) \\ 0, & \text{otherwise} \end{cases} \quad P^n_i = \begin{cases} 1, & \text{if } y^n_{i,(1)} > \hat{y}^{up}(\delta, \sigma) \\ 0, & \text{otherwise} \end{cases}$$

- If $\exists\ P^n_i = 1$, choose the branching dimension n based of the set $\{\sigma^n_1, \dots, \sigma^n_B\}$ that contains the region, σ^n_i, $\sigma^n_i \notin \Sigma^P_k$, with the largest side and $P^n_i = 1$. Set $\Sigma^C_k = \{\sigma^n_i\}$, where σ^n_i is the region in the selected dimension with the largest side and $P^n_i \neq 1$, if there is a tie in side length pick the region with the best sampled value, $y^n_{i,(1)}$.

- Else, if $\exists\ M^n_i = 1$, choose the branching dimension n based of the set $\{\sigma^n_1, \dots, \sigma^n_B\}$ that contains the region, σ^n_i, $\sigma^n_i \notin \Sigma^P_k$, with the largest side and $M^n_i = 1$. Set $\Sigma^C_k = \{\sigma^n_i\}$, where σ^n_i

is the region in the selected dimension with the largest side and $M_i^n = 1$, if there is a tie in side length pick the region with the best sampled value, $y_{i,(1)}^n$.

- Else, choose the region with the largest side, σ_i^n, $\sigma_i^n \notin \Sigma_k^P$, and set $\Sigma_{k-1}^C = \{\sigma_i^n\}$.

STEP 3 Stopping Conditions Check to see if we have reached K_{limit} iterations or if there are no more branchable regions. If so, STOP. Otherwise set $k = k+1$, $\alpha_k = \alpha_{k-1}/B$, $\varepsilon_k = \varepsilon_{k-1}/B$, $\delta_k = \delta_{k-1}/B$ and return to **STEP 1**.

Generally, we set

$$N_k = B \cdot \frac{ln(\alpha_k)}{ln\left(1 - \frac{\varepsilon_k}{v(\sigma)}\right)}$$

to ensure that on average we sample a sufficient number of points to qualify for pruning or maintaining based on the requirements of the PBnB algorithm. Alternatively, the sampling at each iteration can be set to static quantity as is done when solving the paneling problem in Section 3.2.

A brief example is illustrated in Figure 5.5 with three iterations of the PBnB-NP algorithm with $B = 3$ and $\delta = 90\%$. On the first iteration, $k = 1$, we have a $\delta_1 = 90\%$ and select the upper region, σ_1, (in light blue) which is maintained at $\delta_1 = 90\%$ as the "most promising" region. The second iteration attempts to maintain a region within the 90% quantile of the new "most-promising" region, σ_1, and checks to make sure that the region is within the $\delta_2 = 30\%$ of the domain. After maintaining another region σ_2, the next iteration, $k = 2$, the algorithm identifies a "most promising" region within the 90% quantile of the region σ_2, and checks to make sure the region is within $\delta_3 = 10\%$ of the entire domain.

There are several advantages of using this approach. First, since we are using the general sampling and pruning method from PBnB, we can assure that the regions that are incorrectly pruned are no more than ε of the domain volume with certainty $(1-\alpha)^4$ on any iteration. Furthermore the use of the Nested Partition framework prevents the algorithm from developing new sub-regions exponentially. Moreover, if a low number of points are sampled on the early iterations, the algorithm will focus very quickly on sampling within a small region of the domain (only later coming up if that region is pruned), that contains initial good points. This can be helpful if there is an initial point

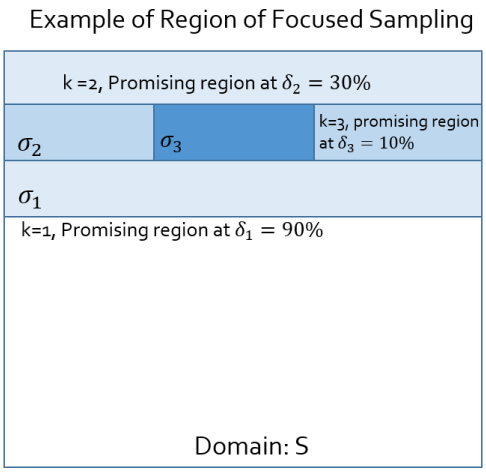

Figure 5.5: An illustration of the NP-PBnB algorithm with three iterations. At $k = 1$ the algorithm attempts to find a promising region that is within the $\delta_1 = 90\%$ of the domain locating σ_1 as the "most promising" region. At $k = 2$ the algorithm attempts to find a region in quantile $\delta_2 = 30\%$ of the domain locating σ_2 as the "most promising" region. Finally, at $k = 3$, the algorithm attempts to find a promising region inside the quantile $\delta_3 = 10\%$ of domain, locating σ_3.

that is particularly promising or if domain knowledge is providing a range of solutions that might be close to an optimum.

Although both the NP-PBnB algorithm and mod-PBnB algorithm provide a way of partially implementing a QAS framework in a practical algorithm, these algorithms cannot fully implement QAS, and therefore do not inherit its ability to efficiently determine near optimal solutions for high dimensional problems. Although these algorithms have been shown to be effective for our particular problems in Chapter 3, a broader perspective on how these methods perform on general optimization problems is needed. In Chapter 6, we explore benchmarking algorithms that explore the effectiveness of these algorithms when tested on general test functions along with other benchmarking efforts to test the effectiveness of optimizers on finding solutions to black-box functions.

Chapter 6

BENCHMARKING EFFICIENT SIMULATION OPTIMIZATION METHODS

The general problem of finding better global optimizers across a broad spectrum of problems is best addressed through optimization benchmarking [67, 42]. Here we look at the performance of a variety of algorithms and compare their performance on common benchmarking problems. The benchmarking effort measures how the optimizers perform relative to the tested problems, in terms of dimensionality, amount of noise, and other problem characteristics.

The basic simulation optimization problem addresses in this chapter is,

$$min_x \; E[g(x,\chi)] \qquad (P2)$$

$$\text{s.t. } x \in S$$

where x is a mixed integer/continuous vector that can be decomposed $x = [x'x'']$ such that $x' \in \mathbb{R}^n$ are the continuous decision variables, and $x'' \in \mathbb{Z}^m$ are the integer decision variables, and S is a domain defined by box constraints such that $l_i \leq x_i \leq u_i$ for an lower and upper bound l_i and u_i for dimension i. The noise is represented by a random variable χ. We denote an optimal solution $x^* \in S$.

Black box optimization algorithms find solutions by sampling the objective function at various points. Generally optimizers run replications in order to estimate the function, as described in Chapter 4. The use of the sample mean can account for noise and prevent the optimization algorithm from tracking objective function values that are dominated by noise. However, sometimes optimizers are run using only single observations or using novel sampling distributions to control for the effects of noise [59]. However, the function evaluations spent performing replications typically comprise a large percentage of the number of function evaluations to control for noise.

As described in Chapter 4 and Chapter 5, the computation of optimization algorithms increse dramatically as the number of dimensions increase. Some optimizers which develop good solutions for low dimensional problems become less well suited for problems that have a large number of

dimensions. Furthermore, the addition of integer constraints can be an issue for many optimization methods [105].

Generally, to demonstrate efficiency optimizers need to be tested against benchmarks with a variety of dimensions and levels of noise in order to demonstrate their utility. Furthermore, there are questions concerning the need for using multiple replications versus other sampling techniques. By varying the noise and dimension on a set of sample problems (whether real simulations or common benchmarking problems), a benchmark effort can compare the efficiency of their performance under a variety of circumstances. We consider this chapter to be a completion of research objective 2(c).

In the following sections we describe several efforts at benchmarking. Section 6.1 describes a benchmarking effort that captures the effect of using a "shrinking ball" technique for noise handling, comparing the estimation technique against the sample mean approach of objective function formulation across a series of 4 selected optimization techniques. In Section 6.2, we extend the benchmark to include analysis of the functions developed in Chapter 5 over the test functions used for benchmarking in Section 6.1.

6.1 Determining the Effect of Replication Techniques on Optimizer Efficiency

Due to the increasing application of stochastic simulations in a variety of practical applications, there is increasing interest in simulation optimization in many research communities. Simulation optimization methods typically focus on black-box problems with no known analytical structure [5]. An alternative method for approximating the fitness of a sampled point for a stochastic black-box function is to examine single observations inside the volume of a hypersphere [13, 6, 51]. The shrinking ball approach uses function values in a neighborhood of each sample point to approximate the fitness at that point. As the sampling proceeds and the number of values in the neighborhood of the point grows, the shrinking ball reduces the average noise of the estimated black-box function. The shrinking ball approach allows for more of the computational budget to be used to explore the domain while accounting for noise through aggregation inside of the ball. The following section overviews results previously published in Proceedings of the Winter Simulation Conference 2017 [59].

A number of papers have measured the relative effectiveness of optimization algorithms for

black-box functions on discrete or continuous domains [73, 28, 60], including applications such as [81, 3, 85, 74]. However, little attention has been given to the relative impact of alternative methods for estimating function response on the performance of optimization algorithms.

This paper describes the application of the shrinking ball approximation method to four different optimizers: Simulated Annealing Pattern Hit-and-Run (SAPHR), Interacting Particle Algorithm Pattern Hit-and-Run (IPAPHR), Particle Swarm Optimization (PSO), and Covariance-Matrix Adaption Evolution Strategy (CMAES). The methods selected constitute a diverse sampling of approaches to black-box optimization that include heuristic approaches (PSO), random search (SAPHR, IPAPHR), and model-based methods (CMAES) as categorized in [5]. Each of the four selected methods are tested with both multiple replication and the shrinking ball approach to estimate the function response.

The test functions are non-convex functions on mixed integer/continuous domains. To compare the selected solvers, we apply each solver to six non-convex functions popular in the benchmark literature [3, 85]. Furthermore, to explore the effect of discretization, we test over a range of dimensions (with both continuous and integer values).

In general, our results suggest, the shrinking ball usually outperforms multiple replications. The performance improvement is more prevalent for low noise, and low dimensions. The percentage of integer variables has little effect. Moreover, this initial benchmark study points to a benefit of using mixed techniques for controlling noise, particularly using single observations in early stages of optimization to improve the exploration of the domain. Some discussion is offered on the relative performance of different shrinking radius rates and opportunities for extending the research.

We use six non-convex test functions from both [3] and [85]. These functions include Ackley's Function (between $[-30, 30]$ on each dimension), Griewank Function (between $[-600, 600]$ on each dimension), Rastrigin Function (between $[-5.12, 5.12]$ on each dimension), Rosenbrock's Function (between $[-2, 2]$ on each dimension), and Sinusoidal Function both centered and shifted (between $[0, 180]$ on each dimension). In order to effectively compare algorithms in a noisy context, we include a random noise factor corresponding to noise of 10% and 20% of the function value respectively. There are six types of test functions with two different dimensions, and three types of integer dimensions, and two levels of noise for a total of $6 \times 3 \times 2 \times 2 = 72$ different trials. Each

optimizer is run with both multiple replications and shrinking ball, for 5000 function evaluations (whether used in replications or otherwise). Each trial is repeated 30 times with different initial sample points.

We tested nine different optimizers, abbreviated as follows ("SAPHR-sb", "SAPHR-mr", "IPAPHR-sb","IPAPHR-mr", "PSO-sb", "PSO-mr", "CMAES-sb","CMAES-mr", "CMAES-nh") to stand for Simulated Annealing with Pattern Hit-and-Run with Shrinking Ball, Simulated Annealing with Pattern Hit-and-Run with multiple replications, Interacting Particle Algorithm Pattern Hit-and-Run with Shrinking Ball, Interacting Particles Algorithm Pattern Hit-and-Run with multiple replications, Particle Swarm Optimizer with Shrinking Ball, Particle Swarm Optimizer with multiple replications, CMAES with Shrinking Ball, CMAES with multiple replications, CMAES with implicit noise handling respectively.

Two comparison metrics are used to track the solver progress. For a given best point x_k^* found at a certain number of function evaluations k, we record two objective values (1. the best estimated function response $\hat{f}(x_k^*)$ averaged over 30 runs, and 2. the "true" value with no noise of the best point found $f_0(x_k^*)$ averaged over 30 runs)

These two measurements provide insight into the progress accomplished by each of the optimization methods by tracking both their approximated and true value of the solutions found. We expect the first value to consistently decrease versus the number of function evaluations, whereas the second may fluctuate due to noise. Differences between the graphs provide initial insight into the affect of noise on the system.

Figure 6.1 illustrates the metrics $\hat{f}(x_k^*)$ and $f_0(x_k^*)$ averaged over 30 runs for the six test functions in ten dimensions, all continuous dimensions, with 10% noise. The experiment generated 12 such arrangements of figures but only one is included in this summary. In Figure 6.1, for each test function, there are two graphs. The left graph plots the approximated function values, $(\hat{f}(x_k^*))$, and the right graph plots "true" value, $(f_0(x_k^*))$.

First, throughout the plots in Figure 6.1 we observe a tendency of methods using shrinking ball approximation to obtain better optimal solutions earlier. This improvement is more pronounced for the approximated function values, $\hat{f}(x_k^*)$, but is still generally present when tracking $f_0(x_k^*)$. The improvement demonstrated by the shrinking ball approach over the multiple replication method is present for all optimizers for some test functions, the Ackley, Griewank, Rastrigin, and Rosenbrock.

For the sinusoidal functions (both shifted and centered) only several optimizers using the shrinking ball approach are superior to the multiple replication approach. By observation we note that, with the exception of the centered sinusoidal function, the CMAES optimizer outperforms the other optimizers (for either of the estimation methods). Additionally, the CMAES and PSO optimizers demonstrate a greater difference between the performance of optimizers using the shrinking ball approach and optimizers using the multiple replication approach.

We can extend the analysis of the performance of these optimizers to different noise and numbers of dimensions based on the descent of the objective function $f_0(x_k^*)$. We prefer optimizers that provide lower points earlier (at fewer function evaluations). This allows us to determine, by visual examination, which optimizer performed better for each of the 72 different trials averaged over 30 runs. For purposes of tabulation, we measure whether optimizer performance using the shrinking ball crosses below the performance of the multiple replication method before 2500 function evaluations. If the shrinking ball performs better than the multiple replication method by this standard, we mark (1) otherwise if the multiple replication method outperforms the shrinking ball approach or if the difference is indistinguishable we mark (0). The next section uses this tabulation metric to explore the effect of the shrinking ball with respect to various aspects of the test function.

The use of the shrinking ball shows a significant effect, across all of the optimizers tested with low noise. We can break down the effectiveness of these optimizers by listing the percent of trials where the shrinking ball approach outperformed that of the multiple replications.

As shown in Table 6.1, with the exception of the sinusoidal functions, the use of the shrinking ball generally improves performance of the optimizers. From a general examination of each optimizer, the shrinking ball almost always shows a much lower estimated value relative to the multiple replication approach. More importantly, this trend carries over to the true function value $f_0(x^*)$, indicating that the use of a shrinking ball approach allows the algorithms to arrive at a lower objective function value earlier. This improved performance is most likely due to the fact that the shrinking ball approach can spend more of its budget on exploring the domain early, where the multiple replication approaches' computational budget is spent on accounting for noise leading to much slower performance of the optimizers tested.

Furthermore, looking at Table 6.1, we can also note that generally the shrinking ball approach

Table 6.1: The percentage over the 12 trials for each test problem where the shrinking ball approach improves the optimizer performance over multiple replications.

	SAPHR-mr	**IPAPHR-mr**	**PSO-mr**	**CMAES-mr**	**CMAES-nh**
Ackley	**50%**	42%	**58%**	**67%**	33%
Griewank	**83%**	**83%**	**100%**	**100%**	25%
Rastrigin	**58%**	**58%**	**83%**	**67%**	8%
Rosenbrock	**100%**	**100%**	**100%**	**100%**	8%
Sinusoidal (shifted)	17%	0%	25%	**58%**	25%
Sinusoidal (centered)	33%	25%	0%	0%	0%

improves performance over PSO-mr and CMAES-mr much more than over the SAPHR-mr or the IPAHR-mr. The CMAES with its own noise-handling, CMAES-nh, outperforms CMAES-sb. Generally both random search approaches (SAPHR and IPAPHR), have very similar performance profiles.

The largest affect on the relative performance of the optimizers is the percentage of noise, η, in the objective function. Here the shift from 10% to 20% noise shows a marked decrease in the effectiveness of solvers that make use of the shrinking ball method to control the noise on the system, as shown in Table 6.2.

Table 6.2: The percentage of 36 trials for each noise level where the shrinking ball approach improves the optimizer performance.

	SAPHR-mr	**IPAPHR-mr**	**PSO-mr**	**CMAES-mr**	**CMAES-nh**
10% - Noise	**81%**	**72%**	**75%**	**83%**	19%
20% - Noise	33%	31%	47%	47%	14%

The effect of the noise on optimizers increases on higher dimensions, but is moderated on test functions with a higher number of integer dimensions, especially for the SAPHR and IPAPHR optimizers. This generally suggests that the effect of noise might be decreased in the integer domains for solvers that are more conservative in their approach but may be less preferable for solvers which focus on optimal solutions early on.

The performance of the algorithms with and without the shrinking ball approach cluster closer together as the number of dimensions increase. Generally at higher dimensions the more ambitious

Table 6.3: The percentage of 36 trials for each number of dimensions where the shrinking ball approach improves the optimizer performance.

	SAPHR-mr	**IPAPHR-mr**	**PSO-mr**	**CMAES-mr**	**CMAES-nh**
10-Dimensional	**67**%	**61**%	**69**%	**69**%	17%
20-Dimensional	47%	42%	**53**%	**61**%	17%

solvers such as the particle swarm with the shrinking ball show greater relative performance than the random search algorithms which show very slow progress inside of the higher dimensions. Nevertheless, there is still a small improvement delivered by using the shrinking ball approach in most trials. However, trials with high noise and high dimension have the multiple replication method outperforming the shrinking ball approach. Another large effect on optimizer performance can be seen by varying the number of integer dimensions in the problems. Generally the number of the integer dimensions improves the overall objective value of the solutions generated by the solvers, causing the optimizers to descend faster towards optimal solutions both in approximation and true value.

Table 6.4: The percentage of 24 for each number of integer dimensions where the shrinking ball approach improves the optimizer performance.

	SAPHR-mr	**IPAPHR-mr**	**PSO-mr**	**CMAES-mr**	**CMAES-nh**
0%-Integer Dimensions	**54**%	46%	**58**%	**63**%	8%
50%-Integer Dimensions	**54**%	50%	**63**%	**67**%	17%
100%-Integer Dimensions	**63**%	**58**%	**63**%	**67**%	25%

However, the effect of the number of integer dimensions on the relative effectiveness of the function response method is not large overall, as shown in Table 6.4. The SAPHR, IPAPHR, and PSO optimizers make the largest improvements with the shrinking ball approach in the domains with a higher number of integer dimensions. We speculate that the pattern hit-and-run sampling techniques allows SAPHR and IPAPHR to more efficiently sample the non-continuous domains which could lead to better use of the algorithm's budget when the shrinking ball method is used.

While the use of the shrinking ball for function response approximation is fairly consistent across the functions with a lower amount of noise and dimension count, most of the improvement

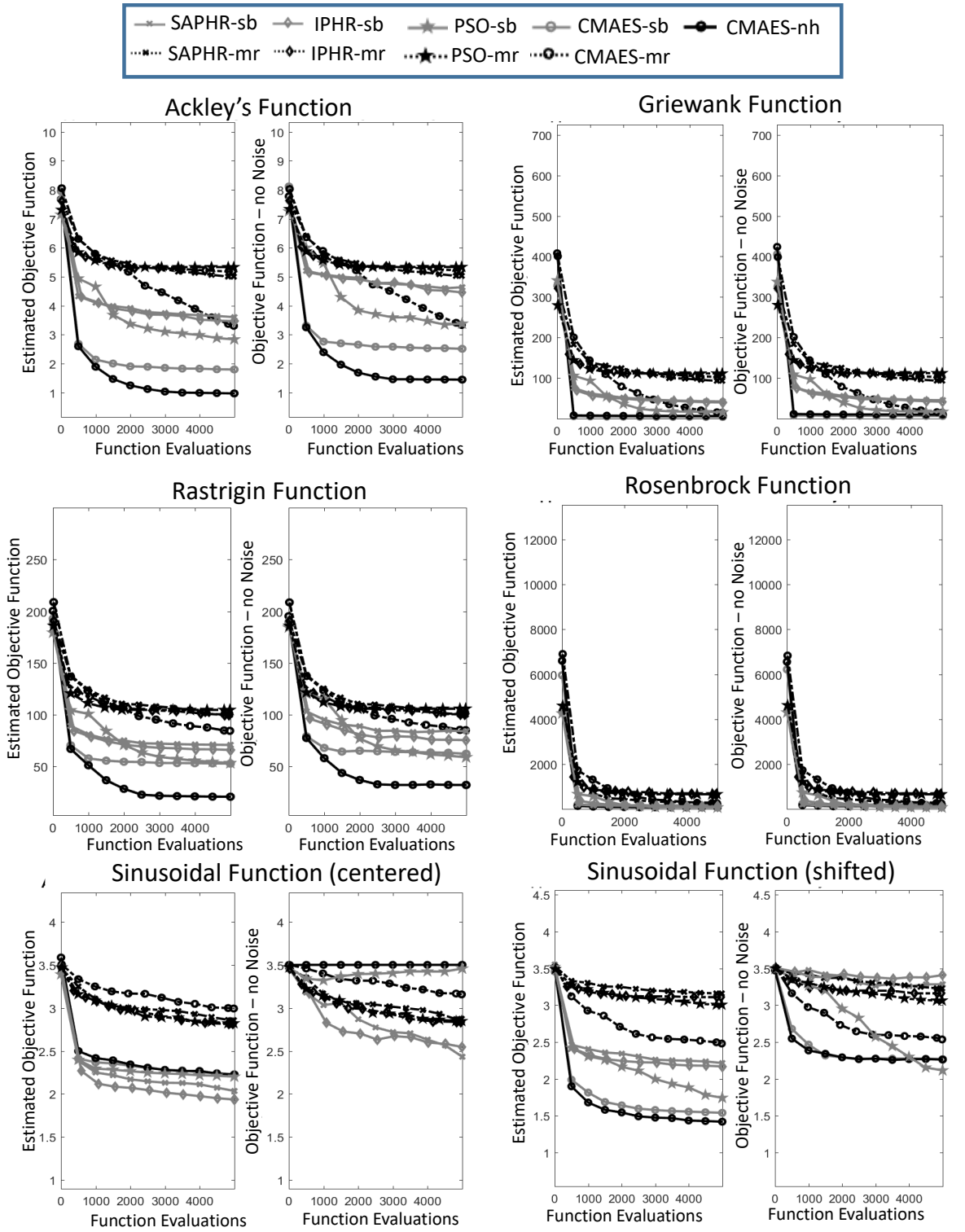

Figure 6.1: The progress of all nine optimizers for the six test functions in 10 dimensions, zero integer dimensions, and 10% noise. In each pair of plots, the left graph plots the estimated function value, and the right graph plots the true function value for each of the optimizers.

demonstrated comes from using single observations to characterize response behavior at a sample point with little influence from other points inside the ball. With the radius set between 1 and 0.5 few sampled points using shrinking ball approximation have previously observed points within the ball $B_{r_k}(x_k)$ for any iteration (with the exception of the CMAES algorithm). This will be explored in future research.

The results point to the effectiveness of taking few replications early in the optimization process under most circumstances. However, the issue of accounting for noise seems to be an issue for the shrinking ball approach in higher dimensions. An additional problem arises in high dimensional domains where a small radius fails to capture enough points to sufficiently account for noise and a large radius captures too many points as it searches the domain. A large amount of this effect might be solved by more ambitious cooling schedules for random search optimizers and lower velocity for the particle swarm. This would allow the various optimization methods to focus sampling in smaller regions which would result in more sampling inside balls of smaller radius. Further research directions might focus on matching a decreasing radius and cooling schedule for high noise and high-dimensional functions in order to better understand the trade-off between radius size and the shrinking rate of the ball.

6.2 Benchmarking Developed Algorithms

In this section we will extend the benchmark analysis to include the problems developed in Chapter 5, we combine this analysis with the shrinking ball formulation to test the optimizer performance under different optimization methods. This gives insight into the potential effectiveness of the new algorithms in addressing general problems across a variety of dimensions and noise level. This analysis also provides insight into the usefulness of algorithms for healthcare decision making going forward, since the need for algorithms with good results on higher dimensions is required to extend the model to more decision variables. Three algorithms are compared to existing algorithms in this section: the modified Probabilistic Branch and Bound algorithm (mod-PBnB, outlined in Section 5.3), the Nested Partition Probabilistic Branch and Bound algorithm (NP-PBnB outlined in Section 5.4), and the Quantile Nested Partition algorithm (QNP, outlined in Section 5.2).

To obtain a standard parameter set fot testing, for mod-PBnB the parameters are set as $\alpha = 0.10$,

$\varepsilon = 0.05$, and $B = 4$. To characterize the lowering quantile level we use two values $\delta_1 = 0.15$ and $\delta_2 = 0.05$ with $K_1' = K_2' = 2$. Running the algorithm for 3 iterations (ending after a maximum number of 5000 function evaluations have been taken). Similarly, the NP-PBnB algorithm is parameterized with $\alpha = 0.10$, $\varepsilon = 0.05$, and $B = 4$. To characterize the starting quantile level we use two value $\delta = 0.15$. Again the algorithm can run 3 iterations before exceeding the 5000 function evaluations. Finally, the Quantile Nested Partition algorithm (QNP), similarly, is run with $M = 4$ (divisions per iteration) $T = 1000$ and $\delta = 0.15$. The QNP algorithm runs for 6 iterations after which we stop sampling at 5000 function evaluations.

As in the previous section, each algorithm is run,with multiple replications ($R = 20$) the shrinking ball with a single observation ($R = 1$). Using both of these settings, we compare the results to each of the 6 test functions, at the various noise levels (10%, 20%), different numbers of dimensions (10, 20), and different number of integer dimensions (0%, 50% , 100%). Running 30 experiments, we can track the progress of the best estimated value and best true value sampled (similar to the previous section). For purposes of comparison, we examine the best sampled "true" value at the $2500th$ function evaluation on each algorithm as a baseline for which to compare the optimizer's effectiveness.

Generally, the new solvers (modPBnB, NP-PBnB, and QNP) show different effects from using the shrinking ball method (comparing the results across all test functions). Here the QNP method shows that the shrinking ball improves the results of the optimization 100% of the time with the shrinking ball improving the performance of mod-PBnB in 44.05% and the performance of NP-PBnB in 52.38% of the trials. This indicates that, for the mod-PBnB and NP-PBnB, the use of the shrinking ball does not significantly impact the performance of the algorithm.

In order to compare the general effectiveness of the new optimizers, we perform a comparison between each of the solvers (with and without shrinking ball estimation), against the other optimizers tested in the previous section. To make the comparison, we examine the values at the 2500 function evaluation for any trial type and adjust it relative to the worst and best objective function value for that specific trial. For instance, given an optimization method s from a set of optimizers $\{1 \ldots, S\}$, and a trial t from a certain set of trials $\{1 \ldots, T\}$, let f_s^t be the average objective function value (without noise) of the optimizer at the 2500 function evaluation, f_{best}^t be the lowest function value at any function evaluation across all optimizers for trial t, and f_{worst}^t be the largest function

value at any function evaluation across all optimizers for trial t. Therefore we denote,

$$\hat{f}_i = \frac{1}{T} \cdot \sum_{t=1}^{T} \frac{|f_s^t - f_{best}^t|}{|f_{worst}^t - f_{best}^t|} \tag{6.1}$$

as an adjusted average performance ranking of an optimization method over a set of trials.

Applying (6.1), we tabulate each of solvers for a different grouping of the trials (divided between test function type, number of dimensions, the percentage of noise, and the number of integer dimensions). Based on this tabulation, we can rank each of the 7 optimizations (4 original plus 3 newly developed) on a scale from 1 (best performing) to 7 (worst performing) using performance metric in (6.1). Based the various ways of grouping the trials, we compare the optimizer effectiveness in Table 6.5 (test-function), Table 6.6 (noise level), Table 6.7 (number of dimensions), and Table 6.8 (number of integer dimensions).

Examining the ranking between different trial functions in Table 6.5 provides insight into the relative effectiveness of the new optimization methods. The comparison between different trial functions shows generally that the mod-PBnB and NP-PBnB algorithms rank 4-5 between most test functions, performing slightly better for the Sinusoidal function and worse for the Ackley and Griewank function. The Quantile Nested Partition algorithm performs worse than all other optimization methods, with the exception of the shifted Sinusoidal function. Generally the comparisons across other groupings provide less insight into the relative performance showing that the new optimizers rank lowest (5, 6, 7), with the modified PBnB algorithm outperforming the simulated annealing on lower dimensions. Generally this suggests that the newly developed solvers only conditionally perform better than classic methods for optimizing higher dimensional problems.

This generally demonstrates that the performance of the new algorithms, while competitive with some existing methods, are not superior across all test functions. Although further tuning of the parameters might be used to improve the performance of the quantile-based algorithms, more research is needed to explore practical methods of focusing sampling to more efficiently find optimal or near optimal solutions.

Table 6.5: Then ranked performance for each optimizer relative to the percentage of integer dimensions. Using adjusted true objective function value at the $2500th$ function evaluation and comparing averaged performance between trials from each solver (both with and without the shrinking ball). Here the best optimizer is marked with a 1 in ascending order to the worst 7.

	Ackley	**Rosenbrock**	**Sinusoidal (shifted)**	**Sinusoidal**	**Griewank**	**Rastrigin**
CMAES	1	2	2	6	1	1
mod-PBnB	5	4	4	3	5	4
NP-PBnB	6	5	5	4	6	5
QNP	7	7	1	7	7	7
SAPHR	3	1	7	1	4	6
IPPHR	4	3	6	2	3	3
PSO	2	6	3	5	2	2

Table 6.6: Then ranked performance for each optimizer relative to the percentage of noised added to then function value. Using adjusted true objective function value at the $2500th$ function evaluation and comparing averaged performance between trials from each solver (both with and without the shrinking ball). Here the best optimizer is marked with a 1 in ascending order to the worst 7.

	10% Noise	**20% Noise**
CMAES	1	1
mod-PBnB	5	5
NP-PBnB	6	6
QNP	7	7
SAPHR	3	2
IPPHR	4	4
PSO	2	3

Table 6.7: Then ranked performance for each optimizer relative to the number of dimensions. Using adjusted true objective function value at the $2500th$ function evaluation and comparing averaged performance between trials from each solver (both with and without the shrinking ball). Here the best optimizer is marked with a 1 in ascending order to the worst 7.

	10 Dimensions	**20 Dimensions**
CMAES	1	2
mod-PBnB	4	6
NP-PBnB	6	5
QNP	7	7
SAPHR	5	1
IPPHR	3	4
PSO	2	3

Table 6.8: Then ranked performance for each optimizer relative to the percentage of integer dimensions. Using adjusted true objective function value at the $2500th$ function evaluation. Here the best optimizer is marked with a 1 in ascending order to the worst 7.

	0% Integer Dimensions	**50% Integer Dimensions**	**100% Integer Dimensions**
CMAES	1	1	1
mod-PBnB	5	5	6
NP-PBnB	6	6	5
QNP	7	7	7
SAPHR	2	3	3
IPPHR	4	4	4
PSO	3	2	2

Chapter 7

SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH

This thesis has addressed two main research objectives. First, to develop models for use in healthcare policy making. Second, to expand global optimization methodology for use in stochastic black box optimization. Our results demonstrate optimal policy recommendations that improve the performance of healthcare decision making in terms of patient-centered metrics. Furthermore, the dissertation has presented new results in theoretical and practical implementation of global optimization algorithms.

Chapter 3 overviews research directed to modeling and decision making for healthcare staffing. We first provide results concerning optimal staffing of specialist care with particular attention to balancing the concerns of a risk averse decision maker against the cost of additional staffing and travel time. We demonstrate the necessity of modeling risk in decision making, and the effects of heavy tail demand distributions on the solution generated by a risk-sensitive optimization method. Further work on paneling strategies models is able to provide insight into optimal primary care staffing arrangement. Modeling the interaction of patients within a primary care system using discrete event simulation methods, global optimization methods can provide insight into optimal ways of staffing. While our model has difficulty due to the high dimensionality of the problem, nevertheless, initial recommendations provided by optimization provide insight into paneling for smaller clinics.

To address the concerns about optimizer performance in high-dimensional domains, Chapter 4 outlines two new adaptive random search frameworks: the Hesitant Adaptive Search with Estimation (HAS-E) and Quantile Adaptive Search (QAS). We prove that HAS-E obtains a key finite time property that shows that the number of iterations needed to achieve a certain value above the minimum with HAS-E increases linearly in terms of the dimension domain. We also prove that the total number of function evaluations, including replications, needed to achieve the result increases as a cubic function of the domain dimension. Similarly, the Quantile Adaptive Search is demonstrated to have finite-time results that demonstrate the number of iterations required to obtain a value above

the maximum increases linearly with domain dimension.

The work done in Chapter 5 explores the implementation of practical algorithms for optimizing high-dimensional problems using partition based methods to confront the issue of high dimensionality. Starting in Section 5.1, we outline an OCBA partition based look-ahead algorithm. Section 5.2 describes an extension of the Nested Partition Framework to sample from certain quantile level sets. While Section 5.3 updates the existing PBnB algorithm to iteratively target lower quantile level-sets, and Section 5.4 incorporates the PBnB algorithm fully into a Nested Partition framework.

Our benchmarking efforts outlined in Chapter 6 consist of a set of experiments that formally measure the performance of optimization algorithms on a variety of different test functions at variable dimensions and noise levels. New algorithms developed in the previous sections are benchmarked across the test problems to demonstrate the effectiveness of the new methods.

7.1 Future Research

Future research can continue along several different directions that will expand both the theoretical contributions as well as the applications to practical healthcare decision making. Further planned papers will explore direct extensions of the existing research contained in this dissertation.

An open area of research is the extension of the QAS framework to problems with estimation, as well as further formalized bounds on the number of replications (total function evaluations) that are required to obtain a value within a range of the optimum. Further expansion of the QAS framework may also examine a way of theoretically setting δ quantile levels in order to obtain a certain probability of implementing the QAS results.

Further research into implementable optimization algorithms may explore provable finite time analysis of the modified Probabilistic Branch and Bound and the Nested Partition Probabilistic Branch and Bound method to specify under what conditions the algorithms should target sampling within certain quantile level sets. Further analysis may also be done to demonstrate under what conditions the practical algorithm can approximately implement QAS and obtain similar finite-time results.

The development of better optimization methods will provide future opportunities for healthcare decision making. The most direct extension for the paneling problem is to extend the model to

include more patient demographic types, locations, and panels to better model the real world data available in a series of medical clinics within a metropolitan area. Although optimization methods are ineffective with the hundreds of decision variables, heuristic rules developed at lower dimensions may be applied to higher dimensional models in order to more effectively determine staffing and paneling strategies.

Further applications of global optimization could be applied to similar staffing a medical resource management problems with similar improvements in policy efficiency. These applications could include models of non-physician care, an analysis of triage staffing, and the organization of hospice and mobile care. The development of improved global optimization algorithms can continue to be useful in finding policy solutions to medical applications and providing decision makers with tools to improve the provision of care to patients.

BIBLIOGRAPHY

[1] Mohamed A. Ahmed and Talal M. Alkhamis. Simulation optimization for an emergency department healthcare unit in Kuwait. *European Journal of Operational Research*, 198(3):936–942, 2009.

[2] Shabbir Ahmed, Ulaş Çakmak, and Alexander Shapiro. Coherent risk measures in inventory problems. *European Journal of Operational Research*, 182(1):226–238, 2007.

[3] M. Montaz Ali, Charoenchai Khompatraporn, and Zelda B. Zabinsky. A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. *Journal of Global Optimization*, 31(4):635–672, 2005.

[4] Mahmoud H Alrefaei and Sigrún Andradóttir. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science*, 45(5):748–764, 1999.

[5] Satyajith Amaran, Nikolaos V. Sahinidis, Bikram Sharda, and Scott J. Bury. Simulation Optimization: A Review of Algorithms and Applications. *Annals of Operations Research*, 240(1):351–380, 2016.

[6] Sigrún Andradóttir and Andrei A Prudius. Adaptive Random Search for Continuous Simulation Optimization. *Naval Research Logistics (NRL)*, 57(6):583–604, 2010.

[7] Hari Balasubramanian, Ritesh Banerjee, Brian Denton, James Naessens, and James Stahl. Improving clinical access and continuity through physician panel redesign. *Journal of General Internal Medicine*, 25(10):1109–1115, 2010.

[8] Hari Balasubramanian, Ritesh Banerjee, Melissa Gregg, and Brian T. Denton. Improving primary care access using simulation optimization. *Proceedings of the 2007 Winter Simulation Conference*, pages 1494–1500, 2007.

[9] Hari Balasubramanian, Sebastian Biehl, Longjie Dai, and Ana Muriel. Dynamic allocation of same-day requests in multi-physician primary care practices in the presence of prescheduled appointments. *Health Care Management Science*, 17(1):31–48, 2014.

[10] Hari Balasubramanian, Ana Muriel, Asli Ozen, Liang Wang, Xiaoling Gao, and Jan Hippchen. Capacity allocation and flexibility in primary care. In *Handbook of healthcare operations management*, pages 205–228. Springer, 2013.

[11] Jerry Banks. *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons, 1998.

[12] Russell R Barton and John S Ivey Jr. Nelder-Mead simplex modifications for simulation optimization. *Management Science*, 42(7):954–973, 1996.

[13] S Baumert and RL Smith. Pure Random Search for Noisy Objective Functions. Technical Report 01-03, University of Michigan, Ann Arbor, 2002.

[14] James C. Benneyan, Hande Musdal, Mehmet Erkan Ceyhan, Brian Shiner, and Bradley V. Watts. Specialty care single and multi-period location allocation models within the Veterans Health Administration. *Socio-Economic Planning Sciences*, 46(2):136–148, jun 2012.

[15] Thomas Bodenheimer and Hoangmai H. Pham. Primary care: current problems and proposed solutions. *Health affairs (Project Hope)*, 29(5):799–805, may 2010.

[16] Justin Boesel, Barry L Nelson, and Nobuaki Ishii. A framework for simulation-optimization software. *IIE Transactions*, 35(3):221–229, 2003.

[17] D. W. Bulger and G. R. Wood. Hesitant adaptive search for global optimisation. *Mathematical Programming*, 81(1):89–102, 1998.

[18] Chun-Hung Chen and Donghai He. Intelligent simulation for alternatives comparison and application to air traffic management. *Journal of Systems Science and Systems Engineering*, 14:37–51, 2005.

[19] Chun Hung Chen, Jianwu Lin, Enver Yücesan, and Stephen E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications*, 10:251–270, 2000.

[20] Hsiao-Chang Chen, Liyi Dai, Chun-Hung Chen, and Enver Yücesan. New development of optimal computing budget allocation for discrete event simulation. In *Proceedings of the Winter Simulation Conference*, pages 334–341. IEEE Computer Society, 1997.

[21] Weiwei Chen, Siyang Gao, Chun Hung Chen, and Leyuan Shi. An optimal sample allocation strategy for partition-based random search. *IEEE Transactions on Automation Science and Engineering*, 11(1):177–186, 2014.

[22] K Coleman, R J Reid, E Johnson, C Hsu, T R Ross, P Fishman, and E Larson. Implications of reassigning patients for the medical home: a case study. *Ann Fam Med*, 8(6):493–498, 2010.

[23] William Jay Conover. Practical nonparametric statistics. 1980.

[24] Michael F Drummond. Resource allocation decisions in health care: a role for quality of life assessments? *Journal of Chronic Diseases*, 40(6):605–616, 1987.

[25] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

[26] T Eldabi, R J Paul, and T Young. Simulation modelling in healthcare: reviewing legacies and investigating futures. *Journal of the Operational Research Society*, 58(2):262–270, 2007.

[27] Melissa L. Finucane, Paul Slovic, C.K. Mertz, James Flynn, and Theresa A. Satterfield. Gender, race, and perceived risk: The 'white male' effect. *Health, Risk & Society*, 2(2):159–172, 2000.

[28] C. A. Floudas and C. E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[29] J. M. Gill. The Effect of Continuity of Care on Emergency Department Use. *Archives of Family Medicine*, 9(4):333–338, 2000.

[30] James M Gill and Arch G Mainous III. The role of provider continuity in preventing hospitalizations. *Archives of family medicine*, 7(4):352–357, 1998.

[31] Lee Goldman, Ralph Freidin, E Francis Cook, John Eigner, and Pamela Grich. A multivariate approach to the prediction of no-show behavior in a primary care center. *Archives of Internal Medicine*, 142(3):563–567, 1982.

[32] Linda V Green. Using Operations Research to Reduce Delays for Healthcare. *Operation Research Informs*, pages 1–16, 2008.

[33] Linda V. Green, Sergei Savin, and Yina Lu. Primary care physician shortages could be eliminated through use of teams, nonphysicians, and electronic communication. *Health Affairs*, 32(1):11–19, 2013.

[34] Linda V. Green, Sergei Savin, and Mark Murray. Providing timely access to care: What is the right patient panel size? *Joint Commission Journal on Quality and Patient Safety*, 33(4):211–218, 2007.

[35] Kevin Grumbach and Thomas Bodenheimer. Can health care teams improve primary care practice? *JAMA : the Journal of the American Medical Association*, 291(10):1246–1251, 2004.

[36] M M Gunal, M Pidd, and M M Günal. Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, 4(1):42–51, 2010.

[37] Murat M Günal and Michael Pidd. Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, 4(1):42–51, 2010.

[38] D. Gupta and L. Wang. Revenue Management for a Primary-Care Clinic in the Presence of Patient Choice. *Operations Research*, 56(3):576–592, 2008.

[39] Diwakar Gupta, Sandra Potthoff, Donald Blowers, John Corlett, and Scott R Terry. Performance metrics for advanced access. *Journal of Healthcare Management*, 51(4):246, 2006.

[40] Nikolaus Hansen and Stefan Kern. Evaluating the cma evolution strategy on multimodal test functions. In *PPSN*, volume 8, pages 282–291. 2004.

[41] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

[42] Eligius M T Hendrix and Algirdas Lančinskas. On Benchmarking Stochastic Global Optimization Algorithms. *Informatica*, 26(4):649–662, 2015.

[43] Tito Homem-de Mello and Güzin Bayraksan. Monte Carlo sampling based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.

[44] Hao Huang. Discrete-event Simulation and Optimization to Improve the Performance of a Healthcare System. *University of Washington Dissertation*, 2016.

[45] Hao Huang and Zelda B Zabinsky. Adaptive probabilistic branch and bound with confidence intervals for level set approximation. In *Proceedings of the 2013 Winter Simulation Conference*, pages 980–991. IEEE Press, 2013.

[46] Y. Huang and D. A. Hanauer. Patient No-Show Predictive Model Development using Multiple Data Sources for an Effective Overbooking Approach. *Applied Clinical Informatics*, 5(3):836–860, 2014.

[47] Grahame A Jastrebski and Dirk V Arnold. Improving evolution strategies through active covariance matrix adaptation. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2814–2821. IEEE, 2006.

[48] Rod Jones. High risk categories and risk pooling in healthcare costs. 18(8), 2012.

[49] JB Jun, Sheldon H Jacobson, and James R Swisher. Application of discrete-event simulation in health care clinics: A survey. *Journal of the operational research society*, 50(2):109–123, 1999.

[50] Maurice G Kendall. *A Course in the Geometry of n Dimensions*. Courier Corporation, 2004.

[51] Seksan Kiatsupaibul, Robert L Smith, and Zelda B Zabinsky. Improving hit-and-run with single observations for continuous simulation optimization. In L. Yilmaz, W.K.V. Chan, I. Moon, T.M.K. Roeder, C. Macal, and M.D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3569–3576, Huntington Beach, CA, 2015. Institute of Electrical and Electronics Engineers, Inc.

[52] Sujin Kim and Dali Zhang. Convergence properties of direct search methods for stochastic optimization. In *Proceedings of the Winter Simulation Conference*, pages 1003–1011. Winter Simulation Conference, 2010.

[53] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[54] David D Linz, Hao Huang, and Zelda B Zabinsky. Partition based optimization for updating sample allocation strategy using lookahead. pages 3577–3588. IEEE, Proceedings of the 2015 Winter Simulation Conference, 2015.

[55] David D Linz, Hao Huang, and Zelda B Zabinsky. A quantile-based nested partition algorithm for black-box functions on a continuous domain. In *Proceedings of the 2016 Winter Simulation Conference*, pages 638–648. IEEE Press, 2016.

[56] David D. Linz, Zelda B. Zabinsky, Joseph A. Heim, and Paul Fishman. Optimal adjusted panel size for balancing patient and physician risk. Philadelphia, PA, 2015. Presented at INFORMS General Conference 2015.

[57] David D. Linz, Zelda B. Zabinsky, Joseph A. Heim, and Paul Fishman. Optimal staffing of patient centered medical homes under conditions of highly variable demand. Nashville, TN, 2015. Presented at INFORMS Healthcare Conference 2015.

[58] David D. Linz, Zelda B Zabinsky, Joseph A Heim, and Paul Fishman. A multi-objective model for optimizing staffing across geographically distributed patient centered medical homes. *Submitted to IIE Transactions on Healthcare Systems Engineering*, 2016.

[59] David D Linz, Zelda B Zabinsky, Seksan Kiatsupaibul, and Robert L Smith. A computational comparison of simulation optimization methods using single observations within a shrinking ball on noisy black-box functions with mixed integer and continuous domains. In *Proceedings of the 2017 Winter Simulation Conference*, pages 2045–2056. IEEE, 2017.

[60] Wang Long-Fei and Shi Le-Yuan. Simulation Optimization: A Review on Theory and Applications. *Acta Automatica Sinica*, 39(11):1957–1968, 2013.

[61] Stephen Mahar, Kurt M. Bretthauer, and Peter A. Salzarulo. Locating specialized service capacity in a multi-hospital network. *European Journal of Operational Research*, 212(3):596–605, aug 2011.

[62] Adele Marshall, Christos Vasilakis, and Elia El-Darzi. Length of stay-based patient flow models: Recent developments and future directions. *Health Care Management Science*, 8(3):213–220, 2005.

[63] Richard T Meenan, Michael J Goodman, Paul A Fishman, Mark C Hornbrook, Maureen C O'Keeffe-Rosetti, and Donald J Bachman. Using risk-adjustment models to identify high-cost risks. *Medical care*, pages 1301–1312, 2003.

[64] Huseyin Onur Mete, Yanfang Shen, Zelda B. Zabinsky, Seksan Kiatsupaibul, and Robert L. Smith. Pattern Discrete and Mixed Hit-and-Run for Global Optimization. *Journal of Global Optimization*, 50(4):597–627, 2011.

[65] Huseyin Onur Mete and Zelda B. Zabinsky. Pattern Hit-and-Run for Sampling Efficiently on Polytopes. *Operations Research Letters*, 40(1):6–11, 2012.

[66] Huseyin Onur Mete and Zelda B Zabinsky. Multiobjective Interacting Particle Algorithm for Global Optimization. *INFORMS Journal on Computing*, 26(3):500–513, 2014.

[67] Stefan More, Jorge; Wild. Benchmarking Derivative-Free Optimization Algorithms. 20(1):172–191, 2009.

[68] Mark Murray and Donald M Berwick. Advanced access: reducing waiting and delays in primary care. *JAMA : the journal of the American Medical Association*, 289(8):1035–1040, 2003.

[69] Mark Murray and Donald M Berwick. Advanced access: reducing waiting and delays in primary care. *Jama*, 289(8):1035–1040, 2003.

[70] Mark Murray, Thomas Bodenheimer, Diane Rittenhouse, and Kevin Grumbach. Improving timely access to primary care: case studies of the advanced access model. *JAMA : the Journal of the American Medical Association*, 289(8):1042–1046, 2003.

[71] National Center for Health Statistics. Health, United States, 2012: With special feature on emergency care. 2013.

[72] Jack Needleman, Peter Buerhaus, Soeren Mattke, Maureen Stewart, and Katya Zelevinsky. Nurse-staffing levels and the quality of care in hospitals. *The New England journal of medicine*, 346(22):1715–1722, 2002.

[73] Arnold Neumaier, Oleg Shcherbina, Waltraud Huyer, and Tamás Vinkó. A Comparison of Complete Global Optimization Solvers. *Mathematical Programming*, 103(2):335–356, 2005.

[74] Anh-tuan Nguyen, Sigrid Reiter, and Philippe Rigo. A Review On Simulation-Based Optimization Methods Applied to Building Performance Analysis. *Applied Energy*, 113:1043–1058, 2014.

[75] Douglas L Nguyen, Ramona S Dejesus, and Mark L Wieland. Missed appointments in resident continuity clinic: patient characteristics and health care outcomes. *Journal of graduate medical education*, 3(3):350–355, 2011.

[76] Vladimir I Norkin, Yuri M Ermoliev, and Andrzej Ruszczyński. On optimal allocation of indivisibles under uncertainty. *Operations Research*, 46(3):381–395, 1998.

[77] World Health Organization. World health statistics, 2011.

[78] Asli Ozen and Hari Balasubramanian. The impact of case mix on timely access to appointments in a primary care group practice. *Health Care Management Science*, 16(2):101–118, 2013.

[79] Giulia Pedrielli and Szu Hui Ng. Kriging-based simulation-optimization: a stochastic recursion perspective. pages 3834–3845, 2015.

[80] Stavros Petrou and Jane Wolstenholme. A review of alternative approaches to healthcare resource allocation. *Pharmacoeconomics*, 18(1):33–43, 2000.

[81] János D. Pintér. Global Optimization: Software, Test Problems, and Applications. *Handbook of Global Optimization*, 2:515–569, 2002.

[82] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[83] Yanto M Prasetio. *Simulation-based Optimization for Complex Stochastic Systems*. 2005.

[84] Abdur Rais and Ana Viana. Operations Research in Healthcare: A Survey. *International Transactions in Operational Research*, 18(1):1–31, Jan 2011.

[85] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[86] L. W. Robinson and R. R. Chen. A Comparison of Traditional and Open-Access Policies for Appointment Scheduling. *Manufacturing & Service Operations Management*, 12(2):330–346, 2010.

[87] H Edwin Romeijn and Robert L Smith. Simulated annealing and adaptive search in global optimization. *Probability in the Engineering and Informational Sciences*, 8(4):571–590, 1994.

[88] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[89] J. W. Saultz and Jennifer Lochner. Interpersonal Continuity of Care and Care Outcomes: A Critical Review. *The Annals of Family Medicine*, 3(2):159–166, 2005.

[90] S Savin. Managing patient appointments in primary care. *Patient Flow: Reducing Delay in Healthcare Delivery*, pages 123–150, 2006.

[91] Robert J Serfling. *Approximation Theorems of Mathematical Statistics*, volume 37. John Wiley & Sons, 1980.

[92] Yanfang Shen. Annealing Adaptive Search With Hit-and-Run Sampling Methods For Global Optimization. *University of Washington Dissertation*, 2005.

[93] Yanfang Shen, Seksan Kiatsupaibul, Zelda B. Zabinsky, and Robert L. Smith. An analytically derived cooling schedule for simulated annealing. *Journal of Global Optimization*, 38(3):333–365, 2007.

[94] Leyuan Shi and Sigurdur Ólafsson. Nested Partitions Optimization. *Tutorials in Operations Research*, (August 2015):1–22, 2007.

[95] Leyuan Shi, Sigurdur Olafsson, et al. *Nested partitions method, theory and applications*. Springer, 2009.

[96] Leyuan; Olafsson Sigurdur Shi. Nested Partition for Global Optimization. *Operations Research*, 11(6):863–888, 2000.

[97] Sue Perrott Siferd and W.C. Benton. Workforce staffing and scheduling: Hospital nursing specific models. *European Journal of Operational Research*, 60(3):233–246, aug 1992.

[98] Vicki Smith-Daniels, Sharon B. Schweikhart, and Dwight E. Smith-Daniels. Capacity Management in Health Care Services: Review and Future Research Directions. *Decision Sciences*, 19(4):889–919, 1988.

[99] Theodore Stefos, James F Burgess, Michael F Mayo-Smith, Kathleen L Frisbee, Henry B Harvey, Laura Lehner, Sophie Lo, and Eileen Moran. The effect of physician panel size on health care outcomes. *Health services management research : an official journal of the Association of University Programs in Health Administration / HSMC, AUPHA*, 24(2):96–105, 2011.

[100] Peter Tulkens Vanberkel. *Interacting hospital departments and uncertain patient flows : theoretical models and applications*. Citeseer, 2011.

[101] Eh Wagner and Katie Coleman. Guiding transformation: how medical practices can become patient-centered medical homes. *The Commonwealth Fund*, (1582):1–25, 2012.

[102] G. R. Wood. The bisection method in higher dimensions. *Mathematical Programming*, 55(1-3):319–337, 1992.

[103] Graham R Wood, Zelda B Zabinsky, and Birna P Kristinsdottir. Hesitant adaptive search: the distribution of the number of iterations to convergence. *Mathematical Programming*, 89(3):479–486, 2001.

[104] Zelda Zabinsky and Hao Huang. A partition-based optimization approach for level set approximation: Probabilistic branch and bound. In *Women in Industrial and Systems Engineering: Key Advances and Perspectives on Emerging Topics (forthcoming)*.

[105] Zelda B Zabinsky. *Stochastic adaptive search for global optimization*, volume 72. Springer Science & Business Media, 2013.

[106] Zelda B. Zabinsky, David Bulger, and Charoenchai Khompatraporn. Stopping and restarting strategy for stochastic sequential search in global optimization. *Journal of Global Optimization*, 46:273–286, 2010.

[107] Zelda B. Zabinsky and Robert L. Smith. Pure adaptive search in global optimization. *Mathematical Programming*, 53(1-3):323–338, 1992.

[108] Zelda B. Zabinsky, Graham R Wood, Mike A. Steel, and WP Baritompa. Pure adaptive search for finite global optimization. *Mathematical Programming*, 69(1-3):443–448, 1995.

[109] Y. Zhang, M. L. Puterman, M. Nelson, and D. Atkins. A Simulation Optimization Approach to Long-Term Care Capacity Planning. *Operations Research*, 60(2):249–261, 2012.

[110] Qipeng P. Zheng, Siqian Shen, and Yuhui Shi. Loss-constrained minimum cost flow under arc failure uncertainty with applications in risk-aware kidney exchange. *IIE Transactions*, 8830(September):1–17, 2015.

Appendix A

OPTIMIZATION ALGORITHMS FOR REFERENCE

A.1 Adaptive Search Algorithms

Adaptive Random Searches exist for solving the standard black-box problem $P0$ in Chapter 4. Common methods including the Pure Adaptive Search (PAS), Hesitant Adaptive Search (HAS), and Annealing Adaptive Search (AAS).

Pure Adaptive Search (PAS) cf. in [107]:

Based on a sampling distribution ρ on S:

- **Step 0:** Initialize X_0 in S according to a probability measure ρ. Set $k = 0$. Set $Y_0 = f(X_0)$.
- **Step 1:** Generate X_{k+1} from the normalized restriction of ρ, on the improving set $S_{k+1} = \{x \in S : f(x) < Y_k\}$ and set $Y_{k+1} = f(X_{k+1})$.
- **Step 2:** If a stopping criterion is met, stop. Otherwise, increment k and return to Step 1.

Annealing Adaptive Search (AAS) cf. [92] :

Based on a Boltzmann sampling distribution and a cooling scheduled based $\tau(Y_k)$ which can be defined as follows:

- **Step 0:** Set $k = 0$. Generate X_0 uniformly distributed on S. Set $Y_0 = f(X_0)$.
- **Step 1:** Generate X_{k+1} from the Boltzmann distribution with parameter T_k over S.
- **Step 2:** If $f(X_{k+1}) < Y_k$, set $Y_{k+1} = f(X_{k+1})$, $Y_{record} = Y_{k+1}$, $X_{record} = X_{k+1}$ and update T_{k+1} such that $T_{k+1} = \tau(Y_{record})$. Otherwise, set $Y_{k+1} = Y_k$ and $T_{k+1} = T_k$.
- **Step 3:** If a stopping criteria is met, stop. Otherwise, increment k and return to Step 1.

A.2 Algorithms for Benchmarking Comparison

A number of optimizers are tested in chapter 6 and are described here.

Simulated Annealing Pattern Hit-and-Run (SAPHR)

Initialization: Set an initial temperature parameter T_0, a random starting point x_0, and set $k = 0$.

Generate New Points: Generate a new candidate point, x'_k, using Pattern Hit-and-Run.

Estimate Function Response: Approximate $\hat{f}(x'_k)$ either by a sample average of multiple replications or through the shrinking ball approach.

Acceptance\Rejection: Calculate an acceptance probability as a function of temperature and the estimated objective function values

$$p_{accept} = e^{((\hat{f}(x_k)-\hat{f}(x'_k))/T_k)}$$

and update x_{k+1} and $\hat{f}(x_{k+1})$ accordingly.

Update Temperature: $T_{k+1} = 500$.

Stopping Condition: If a stopping condition is met, end, otherwise set $k = k+1$ and go to Step 2.

where, the *Pattern Hit-and-Run Generator*, in Step 2, with box search consists of the following steps:

Step 1: Generate a continuous point uniformly distributed in the interior of a box $[-c_1, c_1] \times \ldots [-c_n, c_n]$ for a step-size pattern.

Step 2: Generate a random permutation of n coordinate dimensions.

Step 3: Uniformly select a point on the sample path as the new candidate point from a forward and backward path of permuted directions extending the current point to the boundary of the domain.

These steps are outlined fully in [64, 65].

Interacting Particle Algorithm Pattern Hit-and-Run (IPAPHR)

Initialization: Set an initial temperature parameter T_0 and a set of points as the current points, $x_{0,l}$ where l indexes the number of L particles, set $k = 0$.

Generate New Points: Generate L new candidate points, $x'_{k,l}$, using Pattern Hit-and-Run.

Estimate Function Response: Approximate $\hat{f}(x'_{k,l})$ for each particle $x_{k',l}$, either by a sample average of multiple replications or through the shrinking ball approach.

Acceptance\Rejection: Calculate an acceptance probability as a function of temperature and previous sampled points with their estimated objective function values

$$p_{accept,l} = \frac{G(x_{k,l}, x'_{k,l})}{\sum_{l=1}^{L} G(x_{k,l}, x'_{k,l})} \qquad \text{where } G(x_{k,l}, x'_{k,l}) = e^{((\hat{f}(x_{k,l}) - \hat{f}(x'_{k,l}))/T_k)}$$

and update $x_{k+1,l}$ and $\hat{f}(x_{k+1,l})$ accordingly.

Update Temperature: $T_{k+1} = 500$.

Stopping Condition: If stopping condition is met, end, otherwise set $k = k+1$ and go to Step 2.

Further description can be found in [66].

Particle-Swarm Optimization (PSO)

Initialization: Select locations for a set of L starting points $x_{0,l}$, with initial velocity vector $v_{k,l} = 0$, and $k = 0$.

Generate New Points: Move particles based on velocity to new points such that $x_{k+1,l} = x_{k,l} + v_{k,l}$.

Estimate Function Response: Approximate $\hat{f}(x_{k+1,l})$ for each point either by a sample average of multiple replications or through the shrinking ball approach

Rank Elite Solutions: Rank the estimated function evaluations. Determine the best point visited across all particles x^B_{k+1} and within each particle, $x^{b_l}_{k+1,l}$.

Update Velocity: For each l, determine velocity $v_{k+1,l} = \omega \cdot v_{k,l} + \phi_p \cdot Uniform(0,1) \cdot (x^B_{k+1,l} - x_{k+1,l}) + \phi_g \cdot Uniform(0,1) \cdot (x^{b_l}_{k+1,l} - x_{k+1,l})$.

Stopping Condition: If stopping condition is met, end, otherwise set $k = k+1$ and go to Step 2.

More details can be found in [25, 82].

Covariance-Matrix Adaption Evolution Strategy (CMAES)

Initialization: Set covariance update parameters and initial covariance matrix $C = I$ along with selected mean points m_0, σ_0 and weights w, set $k = 0$.

Generate New Points: Sample L points, $x_{k,l}$, from multivariate normal distribution, $Nor(m_k, \sigma_k \cdot C_k)$, based on mean m_k and covariance matrix update $\sigma_k \cdot C_k$.

Estimate Function Response: Approximate $\hat{f}(x_{k,l})$ for each point either by a sample average of multiple replications or through the shrinking ball approach.

Update Mean Center: Update the mean such that points are weighted by their estimated function value.

Update Covariance: Update the covariance matrix $C_{k+1} = U_{covariance}(C_k, m_k, m_{k+1})$.

Update Variance: Update the variance $\sigma^2_{k+1} = U_{variance}(\sigma_k, m_k, m_{k+1})$.

Stopping Condition: If stopping condition is met, end, otherwise set $k = k+1$ and go to Step 2.

The $U_{covariance}$ and $U_{variance}$ functions are specified in [47]. Further description can be found in [41, 40].

A.3 Probabilistic Branch and Bound (PBnB) [cf [44]]

Step 0. Initialization:

Input user-defined parameters, δ, α, ε, k_b, B, c, and R^0. Also, initialize the maintain, prune, and current subregion collections and iterative counters as $\Sigma_1 = \{S\}$, $\widetilde{\Sigma}_1^C = S$, $\widetilde{\Sigma}_1^M = \phi$, $\widetilde{\Sigma}_1^P = \phi$, $\delta_1 = \delta$, $\alpha_1 = \frac{\alpha}{B}$, $\varepsilon_1 = \frac{\varepsilon}{B}$, $R_0 = R^o$, and $k = 1, k_c = k_b, c_1 = c$.

Step 1. Sample total c_k points in current subregions with updated replication number:

For the current undecided subregion $\widetilde{\Sigma}_k^C$, uniformly sample additional points such that the total number of points in $\widetilde{\Sigma}_k^C$ is c_k. For each subregion $\sigma_i \in \Sigma_k$, denote the sample points as $x_{i,j} \in \sigma_i$, for $j = 1, \ldots, N_k^i$ and $i = 1, \ldots, ||\Sigma_k||$. Note that $\sum_{i=1}^{||\Sigma_k||} N_k^i = c_k$. For notational convenience, let $N_k = c_k$. If $f(x)$ is noisy, PBnB evaluates $g(x, \xi)$ with R_{k-1} replications to estimate mean and variance of each sample. Specifically, for each $x_{i,j} \in \sigma_i$, $j = 1, \ldots, N_k^i$ and $i = 1, \ldots, ||\Sigma_k||$, perform R_{k-1} replications of $g(x, \xi_x^r)$, and evaluate the sample mean and sample variance,

$$\hat{f}(x_{i,j}) = \frac{\sum_{r=1}^{R_{k-1}} g(x_{i,j}, \xi_x^r)}{R_{k-1}} \text{ and } S_{\hat{f}}^2(x_{i,j}) = \frac{1}{(R_{k-1}-1)} \sum_{r=1}^{R_{k-1}} (g(x_{i,j}, \xi_x^r) - \hat{f}(x_{i,j}))^2. \qquad \text{(A.1)}$$

Step 2. Order samples in each subregion and over all subregions by estimated function values:

For each subregion i, $i = 1, \ldots, ||\Sigma_k||$, order the sampled points, $x_{i,(1)}, \ldots, x_{i,(N_k^i)}$, by their estimated function value so that

$$\hat{f}(x_{i,(1)}) \leq \hat{f}(x_{i,(2)}) \leq \cdots \leq \hat{f}(x_{i,(N_k^i)}).$$

Similarly, order all sampled points, $z_{(1)}, \ldots, z_{(N_k)}$, in all current subregions in Σ_k by their

function values, so that

$$\hat{f}(z_{(1)}) \leq \hat{f}(z_{(2)}) \leq \cdots \leq \hat{f}(z_{(N_k)}).$$

If $f(x)$ is noisy, we need to check the ordering with further replications calculated as follows. (2.A) Calculate the differences between ordered samples, let $d_{i,j} = \hat{f}(x_{i,(j+1)}) - \hat{f}(x_{i,(j)})$, where $i = 1,\ldots,||\Sigma_k||$ and $j = 1,\ldots,N_k^i - 1$. Determine $d^* = \min_{i=1,\ldots,||\Sigma_k||, j=1,\ldots,N_k^i-1} d_{i,j}$ and $S^{*2} = \max_{i=1,\ldots,||\Sigma_k||, j=1,\ldots,N_k^i} S^2_{\hat{f}}(x_{i,(j)})$. (2.B) Calculate the updated replication number $R_k = \max\left\{R_{k-1}, \left(\frac{z_{\alpha_k/2} S^*}{d^*/2}\right)^2\right\}$, where $z_{\alpha_k/2}$ is the $1-\alpha_k/2$ quantile of the standard normal distribution. Perform $R_k - R_{k-1}$ more replications for each sample point. Re-estimate the performance of each sample point with R_k replications by $\hat{f}(x_{i,j}) = \frac{\sum_{r=1}^{R_k} g(x_{i,j}, \xi^r_{x_{i,j}})}{R_k}$. Within each subregion $\sigma_i \in \Sigma_k$, rank all the sample points $x_{i,j}$ as $x_{i,(j)}$ representing the jth best point in subregion, according to the estimated function value, and also update the entire order of all current samples with updated replications, so that

$$\hat{f}(x_{i,(1)}) \leq \hat{f}(x_{i,(2)}) \leq \cdots \leq \hat{f}(x_{i,(N_k^i)}), \text{ and } \hat{f}(z_{(1)}) \leq \hat{f}(z_{(2)}) \leq \cdots \leq \hat{f}(z_{(N_k)}).$$

Step 3. Build confidence interval for $y(\delta, S)$:

To build the confidence interval of quantile $y(\delta, S)$, first, calculate the lower and upper bounds of δ_k as

$$\delta_{kl} = \delta_k - \frac{v(\widetilde{\Sigma}_k^P)\varepsilon}{v(S)v(\widetilde{\Sigma}_k^C)} \text{ and } \delta_{ku} = \delta_k + \frac{v(\widetilde{\Sigma}_k^M)\varepsilon}{v(S)v(\widetilde{\Sigma}_k^C)}. \tag{A.2}$$

Then calculate the confidence interval lower bound $CI_l = \hat{f}(z_{(r)})$ and the upper bound $CI_u = \hat{f}(z_{(s)})$, where r and s are selected by

$$\max r : \sum_{i=0}^{r-1} \binom{N_k}{i} (\delta_{kl})^i (1-\delta_{kl})^{N_k - i} \leq \frac{\alpha_k}{2} \text{ and} \tag{A.3}$$

$$\min s : \sum_{i=0}^{s-1} \binom{N_k}{i} (\delta_{ku})^i (1-\delta_{ku})^{N_k - i} \geq 1 - \frac{\alpha_k}{2}. \tag{A.4}$$

Step 4. Find elite and worst subregions, and further sample with updated replications: Step 4 identifies the indices of elite and worst subregions as e and w, representing the subregions

that are likely to be maintained or pruned. The sets e and w are defined with the quantile confidence interval as

$$e = \{i | \hat{f}(x_{i,(N_k^i)}) < CI_l, \text{ for } i \in 1, \ldots, ||\Sigma_k||\} \tag{A.5}$$

$$w = \{i | \hat{f}(x_{i,(1)}) > CI_u, \text{ for } i \in 1, \ldots, ||\Sigma_k||\}. \tag{A.6}$$

Statistically confirm maintaining and pruning for each elite or worst subregion by sampling points up to N_k^n, where

$$N_k^n = \left\lceil \frac{\ln \alpha_k}{\ln (1 - \frac{\varepsilon}{v(S)})} \right\rceil, \text{for all } i \in \{e \cup w\}. \tag{A.7}$$

For each new sample, perform R_k replications, and evaluate the sample mean and sample variances as in (A.1). Reorder the sampled points in each subregion σ_i, and update d^* and S^{*2} as in (2.A). As in (2.B), calculate the updated replication number $R_k^n = \max\left\{R_k, \left(\frac{z_{\alpha_k/2} S^*}{d^*/2}\right)^2\right\}$, where $z_{\alpha_k/2}$ is the $1 - \alpha_k/2$ quantile of the standard normal distribution. Perform $R_k^n - R_k$ more replications for each new sample point. Update $\hat{f}(x_{i,(1)}) = \min_{x_{i,j} \in \sigma_i} \hat{f}(x_{i,j})$ and $\hat{f}(x_{i,(N_k^i)}) = \max_{x_{i,j} \in \sigma_i} \hat{f}(x_{i,j})$ for $i \in \{e \cup w\}$.

Step 5. Maintain, Prune, and Branch:

Update the maintaining indicator functions M_i, for $i \in e$, and the pruning indicator functions P_i, for $i \in w$, as

$$M_i = \begin{cases} 1, \text{ if } \hat{f}(x_{i,(N_k^i)}) < CI_l \\ 0, \text{ otherwise} \end{cases} \quad \text{and } P_i = \begin{cases} 1, \text{ if } \hat{f}(x_{i,(1)}) > CI_u \\ 0, \text{ otherwise.} \end{cases} \tag{A.8}$$

Update the maintained set $\widetilde{\Sigma}_{k+1}^M$ and the pruned set $\widetilde{\Sigma}_{k+1}^P$ as

$$\widetilde{\Sigma}_{k+1}^M = \widetilde{\Sigma}_k^M \bigcup_{i \in e: M_i = 1} \sigma_i \quad \text{and} \quad \widetilde{\Sigma}_{k+1}^P = \widetilde{\Sigma}_k^P \bigcup_{i \in w: P_i = 1} \sigma_i,$$

and branch the remaining current subregions in the following manner. (1) If all subregions $\sigma_i \in \Sigma_k$ are not branchable, terminate the algorithm. Else, if σ_i is branchable, and if σ_i,

$i = 1, \ldots, ||\Sigma_k||$, has not been maintained or pruned, then partition σ_i to $\bar{\sigma}_i^1, \ldots, \bar{\sigma}_i^B$ and update the current set of subregions

$$\Sigma_{k+1}^C = \{\bar{\sigma}_i^j : \forall i \text{ to be branched}, j = 1, \ldots, B\} \text{ and } \widetilde{\Sigma}_{k+1}^C = \bigcup_{i \text{ to be branched}} \left(\bigcup_{j=1}^{B} \bar{\sigma}_i^j \right).$$

Determine δ_{k+1} by

$$\delta_{k+1} = \frac{\delta_k v(\widetilde{\Sigma}_k^C) - \sum_{i:M_i=1} v(\sigma_i)}{v(\widetilde{\Sigma}_k^C) - \sum_{i:P_i=1} v(\sigma_i) - \sum_{i:M_i=1} v(\sigma_i)}. \tag{A.9}$$

Set

$$k_c = \begin{cases} k_c + 1 & , \text{ if } \sum_{i \in e} M_i + \sum_{i \in w} P_i = 0 \\ 0 & , \text{ otherwise.} \end{cases} \tag{A.10}$$

Set $\alpha_{k+1} = \frac{\alpha_k}{B}$, $\varepsilon_{k+1} = \frac{\varepsilon_k}{B}$, $c_{k+1} = c_k + c$, and increment $k \leftarrow k+1$. (2) If $k_c \geq k_b$, set $k_c = 1$ and go to Step 1. (3) If $k_c < k_b$, go to Step 4.

Appendix B

PROOFS FOR THEOREMS

B.1 Additional Lemma with Proof

Here we repeat Lemma 30 from [92], which is used in Theorem 2 and Theorem 6.

Lemma 11. ***(cf. [92])*** *Let $\bar{Y}_k^A, k = 1,2,\ldots$ and $\bar{Y}_k^B, k = 1,2,\ldots$ be two sequences of objective function values generated by algorithms A and B respectively for solving an optimization problem, such that $\bar{Y}_{k+1}^A \leq \bar{Y}_k^A$ and $\bar{Y}_{k+1}^B \leq \bar{Y}_k^B$ for $k = 1,2,\ldots$. For $y_* < y, z \leq y^*$ and $k = 0,1,\ldots$, if*

1. $P(\bar{Y}_{k+1}^A \leq y | \bar{Y}_k^A = z) \geq P(\bar{Y}_{k+1}^B \leq y | \bar{Y}_k^B = z)$
2. $P(\bar{Y}_{k+1}^A \leq y | \bar{Y}_k^A = z)$ *is non-increasing in z, and*
3. $P(\bar{Y}_0^A \leq y) \geq P(\bar{Y}_0^B \leq y)$

then $P(Y_k^A \leq y) \geq P(Y_k^B \leq y)$ for $k = 0,1,\ldots$ and $y_* \leq y \leq y^*$.

Proof of Lemma 11 included in [92].

B.2 Hesitant Adaptive Search with Estimation

Theorem 1 Given a function, f, and a point x_k in the domain with a value $y_k = f(x_k)$, and $\hat{y_k}^{high}$ estimated with $R_k = R$ replications, for any given value $0 < q < 1$ and $\varepsilon > 0$ such that $y_k \geq y_* + \varepsilon$ if

$$R \geq \left(\frac{\sqrt[n]{q} \cdot 2 \cdot z_{\alpha/2} \cdot \sigma}{(1 - \sqrt[n]{q}) \cdot r_{y_* + \varepsilon} \cdot \mathscr{K}_q} \right)^2$$

then:

$$\frac{\nu(S_{y_k})}{\nu(S_{\hat{y_k}^{high}})} \geq q$$

when $y_k \leq \hat{y}_k^{high} \leq y_k + \frac{2 \cdot \sigma \cdot z_{\alpha/2}}{\sqrt{R}}$ (which occurs with probability $(1 - \alpha)$).

Proof of Theorem 1

For any value y such that $y_* + \varepsilon < y_k < y^*$, we start by defining an n-ball $\mathscr{B}_{y_k}$ as the largest n-ball, centered at x_* such that $\mathscr{B}_{y_k} \subset S_y$ and let r_{y_k} be its radius. We note that $0 < \nu(\mathscr{B}_{y_k}) < \nu(S_{y_k})$. For any value $\hat{y}_k^{high}$, we define $\mathscr{B}'_{\hat{y}_k^{high}}$ as the smallest n-ball centered at x_* such that $S_{\hat{y}_k^{high}} \subset \mathscr{B}'_{\hat{y}_k^{high}}$ and let $r_{\hat{y}_k^{high}}$ be its radius. We also define the slope of the cone between the two balls $\mathscr{K}_{cone} = \frac{\hat{y}_k^{high} - y_k}{r_{\hat{y}_k^{high}} - r_{y_k}}$, and write $r_{\hat{y}_k^{high}} = r_{y_K} + (\hat{y}_k^{high} - y_k)/\mathscr{K}_{cone}$ by the definition of a cone, as shown in Figure B.1.

To demonstrate when (4.9) holds, we two examine cases. First, if $\hat{y}_k^{high} - y_k < \kappa_q$ then $\frac{\nu(S_{y_k})}{\nu(S_{\hat{y}_k^{high}})} \geq q$ by definition. Second, consider $\hat{y}_k^{high} - y_k > \kappa_q$. Since $\hat{y}_k^{high} - y_k > \kappa_q$, the numerator of $\mathscr{K}_{cone}$ is greater than the numerator of $\mathscr{K}_q$, and, since $d > r_{\hat{y}_k^{high}} - r_{y_k}$ by definition of the diameter. Therefore in combination $\mathscr{K}_{cone} > \mathscr{K}_q$. Note that $\mathscr{K}_q$ is independent of the value y_k.

If we define $\mathscr{B}^{large}$ as an n-ball centered at x_* with radius r_{large}, where $r_{large} = r_{y_K} + (\hat{y}_k^{high} - y_k)/\mathscr{K}_q$. Here we see that $r_{\hat{y}_k^{high}} \leq r_{large}$ since $\mathscr{K}_q \leq \mathscr{K}_{cone}$. Therefore $S_{\hat{y}_k^{high}} \subset \mathscr{B}'_{\hat{y}_k^{high}} \subset \mathscr{B}^{large}$, as shown in Figure B.1.

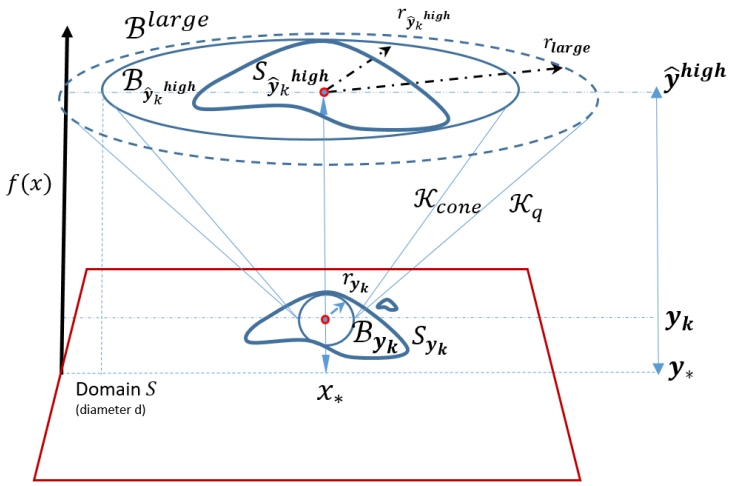

Figure B.1: An illustration of the largest n-ball inscribed in S_{y_K} ($\mathscr{B}_{y_k}$), the smallest n-ball inscribing $S_{\hat{y}_k^{high}}$ ($\mathscr{B}^{large}_{\hat{y}_k^{high}}$), and a larger ball defined by the slope $\mathscr{K}_q$ ($\mathscr{B}^{large}$).

Based on these definitions a lower bound on the ratios of volumes is constructed based on multi-

dimensional geometry theorems [50, 107]:

$$\frac{\nu(S_{y_k})}{\nu(S_{\hat{y_k}^{high}})} \geq \frac{\nu(\mathscr{B}_{y_k})}{\nu(\mathscr{B}_{\hat{y}_k^{high}})} \geq \frac{\nu(\mathscr{B}_{y_k})}{\nu(\mathscr{B}^{large})} = \left(\frac{r_{y_k}}{r_{y_k} + \frac{\hat{y}_k^{high} - y_k}{\mathscr{K}_q}} \right)^n .$$

based on the standard normal estimator, $\hat{y}_k^{high} \sim Nor(y_k + \frac{\sigma \cdot z_{\alpha/2}}{\sqrt{R}}, \frac{\sigma}{\sqrt{R}})$, then the event $\hat{y}_k^{high} - y_k \leq \frac{2 \cdot \sigma \cdot z_{\alpha/2}}{\sqrt{R}}$ provides a lowerbound:

$$\frac{\nu(S_{y_k})}{\nu(S_{\hat{y_k}^{high}})} \geq \left(\frac{r_{y_k}}{r_{y_k} + \frac{2 \cdot \sigma \cdot z_{\alpha}}{\mathscr{K}_q \sqrt{R}}} \right)^n .$$

Setting q as less than the developed lower-bound, we can find R to satisfy

$$\left(\frac{r_{y_k}}{r_{y_k} + \frac{2 \cdot \sigma \cdot z_{\alpha}}{\mathscr{K}_q \sqrt{R}}} \right)^n \geq q$$

taking a positive exponent of the two positive expressions,

$$\left(\frac{r_{y_k}}{r_{y_k} + \frac{2 \cdot \sigma \cdot z_{\alpha}}{\mathscr{K}_q \sqrt{R}}} \right) \geq \sqrt[n]{q}$$

multiplying by a positive term,

$$r_{y_k} \geq \sqrt[n]{q} \cdot \left(r_{y_k} + \frac{2 \cdot \sigma \cdot z_{\alpha}}{\mathscr{K}_q \sqrt{R}} \right)$$

subtracting,

$$r_{y_k} - \sqrt[n]{q} \cdot r_{y_k} \geq \sqrt[n]{q} \cdot \frac{2 \cdot \sigma \cdot z_{\alpha}}{\mathscr{K}_q \sqrt{R}}$$

dividing by a positive number

$$\sqrt{R} \geq \sqrt[n]{q} \cdot \frac{2 \cdot \sigma \cdot z_{\alpha}}{\left(\mathscr{K}_q \sqrt{R} \right) \left((r_{y_k}) - \sqrt[n]{q} \cdot r_{y_k} \right)}$$

and taking a positive exponent of the two positive expressions

$$R \geq \left(\frac{2 \cdot \sqrt[n]{q} \cdot z_{\alpha/2} \cdot \sigma}{(1 - \sqrt[n]{q}) \cdot r_{y_k} \cdot \mathscr{K}_q} \right)^2 \tag{B.1}$$

So (4.9) holds if $R \geq \left(\frac{2 \cdot \sqrt[n]{q} \cdot z_{\alpha/2} \cdot \sigma}{(1 - \sqrt[n]{q}) \cdot r_{y_k} \cdot \mathscr{K}_q} \right)^2$. Finally for any $y_k > y_* + \varepsilon$ then $r_{y_k} > r_{y_* + \varepsilon}$ therefore, the lowerbound will hold if

$$R \geq \left(\frac{2 \cdot \sqrt[n]{q} \cdot z_{\alpha/2} \cdot \sigma}{(1 - \sqrt[n]{q}) \cdot r_{y_{y_* + \varepsilon}} \cdot \mathscr{K}_q} \right)^2$$

, which proves the Theorem 1 holds when $\bar{y} \leq \hat{y}_k^{high} \leq \bar{y} + \frac{2 \cdot \sigma \cdot z_{\alpha/2}}{\sqrt{R}}$ which occurs with probability $(1 - \alpha)$. $\square$

Theorem 2 Consider problem (P1). Let $\bar{Y}_k^{HASE}$ be the best sampled value on the kth iteration of the HAS-E algorithm with sampling distribution ζ and constant bettering probability γ. Let $\bar{Y}_k^{HAS1}$ be the best sampled value of HAS1, the special case of the HAS algorithm with bettering probability $(1 - \alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}} \right)^2 \cdot q \cdot \gamma$ and uniform sampling distribution. $\bar{Y}_k^{HASE}$ stochastically dominates $\bar{Y}_k^{HAS1}$, that is:

$$P(\bar{Y}_k^{HASE} \leq y) \geq P(\bar{Y}_k^{HAS1} \leq y) \text{ for } k = 0, 1, \ldots$$

where $y_* < y \leq y^*$.

Proof of Theorem: 2 Based on Lemma 11 in [92] , if the following conditions hold for $y_* \leq y, z \leq y^*$ and $k = 0, 1, \ldots,$

(I) $P(\bar{Y}_{k+1}^{HASE} \leq y | \bar{Y}_k^{HASE} = z) \geq P(\bar{Y}_{k+1}^{HAS1} \leq y | \bar{Y}_k^{HAS1} = z)$

(II) $P(\bar{Y}_{k+1}^{HASE} \leq y | \bar{Y}_k^{HASE} = z)$ is non-increasing in z, and

(III) $P(\bar{Y}_0^{HASE} \leq y) \geq P(\bar{Y}_0^{HAS1} \leq y)$

then $P(\bar{Y}_k^{HASE} \leq y) \geq P(\bar{Y}_k^{HAS1} \leq y)$ for $k = 0, 1, 2, \ldots$ for $y < z$.

The first step is to prove (I). Let $z = \bar{y}_k$ to reflect the notation in HAS-E on the kth iteration. If $y > \bar{y}_k$, (I) is true trivially (since the conditional probability equals one on both sides). We assume, WLOG, that $y \leq \bar{y}_k$ (since $\bar{y}_k$ is the best sampled value) and re-write the left-hand side of the

expression in (I), replacing z with $\bar{y}_k$ as,

$$\begin{aligned} P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} \geq \bar{y}_k) &= P\left(\{X_{k+1}^{HASE} \in S_y\} | Y_k^{HASE} = \bar{y}_k\right) \\ &= P\left(\{X_{k+1}^{HASE} \in S_y\} \bigcap \{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} \bigcap \{X_{k+1}^{HASE} \text{“ betters” }\} | Y_k^{HASE} = \bar{y}_k\right) \\ &\quad + P\left(\{X_{k+1}^{HASE} \in S_y\} \bigcap \{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} \bigcap \{X_{k+1}^{HASE} \text{ “does not better” }\} | Y_k^{HASE} = \bar{y}_k\right) \end{aligned} \tag{B.2}$$

where the event “betters” is the event that $X_{k+1} \neq X_k$ and $f(X_{k+1}) < f(X_k) = \bar{y}_k$ (which occurs with probability γ) and the event “does not better” is $X_{k+1} = X_k$, and since all terms are positive,

$$\geq P\left(\left\{X_{k+1}^{HASE} \in S_y\right\} \bigcap \left\{X_{k+1}^{HASE} \in S_{\bar{y}_k}\right\} \bigcap \left\{X_{k+1}^{HASE} \text{ “betters” }\right\} | \left\{Y_k^{HASE} = \bar{y}_k\right\}\right). \tag{B.3}$$

Re-write the right-hand side of (B.3) as a product of three terms (e.g., $P(A \bigcap B \bigcap C|D) = P(A|B \bigcap C \bigcap D) \cdot P(B|C \bigcap D) \cdot P(C|D)$), yielding:

$$\begin{aligned} &P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k) \\ &\quad \geq P\left(\{X_{k+1}^{HASE} \in S_y\} | \{X_{k+1}^{HASE} \in S_{\bar{y}_k}\}, \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \\ &\qquad \cdot P\left(\{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} | \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \cdot P\left(\{X_{k+1}^{HASE} \text{ “betters” }\} | Y_k^{HASE} = \bar{y}_k\right). \end{aligned} \tag{B.4}$$

The third term of (B.4) equals the constant bettering probability γ used in the definition of HAS-E, that is,

$$P\left(\{X_{k+1}^{HASE} \text{ “betters” }\} | Y_k^{HASE} = \bar{y}_k\right) = \gamma. \tag{B.5}$$

Furthermore, a lower bound on the second term is:

$$\begin{aligned} &P\left(\{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} | \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \\ &= P\left(\{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} \bigcap \bar{y}_k + \frac{2\sigma \cdot z_{\alpha/2}}{\sqrt{R}} \leq \bar{y}_k^{high} | \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \\ &\quad + P\left(\{X_{k+1}^{HASE} \in S_{\bar{y}_k}\} \bigcap \bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k + \frac{2\sigma \cdot z_{\alpha/2}}{\sqrt{R}} | \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \end{aligned}$$

$$+P\left(\{X_{k+1}^{HASE}\in S_{\bar{y}_k}\}\bigcap \bar{y}_k > \bar{y}_k^{high}|\{X_{k+1}^{HASE}\text{ ``betters'' }\},Y_k^{HASE}=\bar{y}_k\right)$$

and since all probability terms are positive, we can construct a lower bound by dropping the first and third term of the previous expression,

$$\geq P\left(\{X_{k+1}^{HASE}\in S_{\bar{y}_k}\}\bigcap \bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}}|\{X_{k+1}^{HASE}\text{ ``betters'' }\},Y_k^{HASE}=\bar{y}_k\right)$$

and expressing the intersection as the product of the probability of a conditional statement and a condition,

$$\begin{aligned}=P&\left(\{X_{k+1}^{HASE}\in S_{\bar{y}_k}\}|\bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}},\{X_{k+1}^{HASE}\text{ ``betters'' }\},Y_k^{HASE}=\bar{y}_k\right)\\ &\cdot P\left(\bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}}|\{X_{k+1}^{HASE}\text{ ``betters'' }\},Y_k^{HASE}=\bar{y}_k\right)\end{aligned}$$

and since $P(\bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}}|\{X_{k+1}^{HASE}\text{``betters'' }\},Y_k^{HASE}=\bar{y}_k)=(1-\alpha)$, we get,

$$\begin{aligned}=P&\left(\{X_{k+1}^{HASE}\in S_{\bar{y}_k}\}|\bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}},\{X_{k+1}^{HASE}\text{ ``betters'' }\},Y_k^{HASE}=\bar{y}_k\right)\\ &\cdot(1-\alpha)\end{aligned}$$

since the condition $\bar{y}_k \leq \bar{y}_k^{high} \leq \bar{y}_k+\frac{2\sigma\cdot z_{\alpha/2}}{\sqrt{R}}$ holds for the new probability statements, we use the lower bound developed by Theorem 1, $\frac{\nu(S_{\bar{y}_k})}{\nu(S_{\bar{y}_k^{high}})}\geq q$ combining this with the upper and lower bounds for the sampling distribution density,

$$\begin{aligned}&=\frac{\int_{S_{\bar{y}_k}}\zeta(x)\cdot dx}{\int_{S_{\bar{y}_k^{high}}}\zeta(x)\cdot dx}\cdot(1-\alpha)\\ &\geq\frac{\zeta_{low}\cdot\int_{S_{\bar{y}_k}}1\cdot dx}{\zeta_{high}\cdot\int_{S_{\bar{y}_k^{high}}}1\cdot dx}\cdot(1-\alpha)\\ &=\frac{\zeta_{low}\cdot\nu(S_{\bar{y}_k})}{\zeta_{high}\cdot\nu(S_{\bar{y}_k^{high}})}\cdot(1-\alpha)\\ &\geq\frac{\zeta_{low}}{\zeta_{high}}\cdot q\cdot(1-\alpha).\end{aligned}\tag{B.6}$$

Finally, expand the first term in (B.4):

$$
\begin{aligned}
&P\left(X_{k+1}^{HASE} \in S_y | \{X_{k+1}^{HASE} \in S_{\bar{y}_k}\}, \{X_{k+1}^{HASE} \text{ “betters” }\}, Y_k^{HASE} = \bar{y}_k\right) \\
&= \frac{\int_{S_y} \zeta(x) \cdot dx}{\int_{S_{\bar{y}_k}} \zeta(x) \cdot dx} \\
&\geq \frac{\zeta_{low}}{\zeta_{high}} \cdot \frac{\int_{S_y} 1 \cdot dx}{\int_{S_{\bar{y}_k}} 1 \cdot dx} \\
&= \frac{\zeta_{low}}{\zeta_{high}} \cdot \frac{\nu(S_y)}{\nu(S_{\bar{y}_k})}
\end{aligned}
$$

and since HAS1 samples uniformly,

$$
= \frac{\zeta_{low}}{\zeta_{high}} \cdot \frac{\nu(S_y)}{\nu(S_{\bar{y}_k})} = \frac{\zeta_{low}}{\zeta_{high}} \cdot P\left(\{X_{k+1}^{HAS1} \in S_y\} | \{X_{k+1}^{HAS1} \in S_{\bar{y}_k}\}, Y_k^{HAS1} = \bar{y}_k\right). \tag{B.7}
$$

Combining the lower bounds on (B.5) - (B.7) with (B.4), we get:

$$
\begin{aligned}
&P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k) \\
&\geq \left(\frac{\zeta_{low}}{\zeta_{high}}\right) \cdot P\left(\{X_{k+1}^{HAS1} \in S_y\} | \{X_{k+1}^{HAS1} \in S_{\bar{y}_k}\}, Y_k^{HAS1} = \bar{y}_k\right) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right) \cdot q \cdot \gamma \cdot (1-\alpha)
\end{aligned}
$$

and since the bettering probability of HAS1 equals $(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma$, and since $P\left(\{X_{k+1}^{HAS1} \in S_y\} | \{X_{k+1}^{HAS1} \notin S_{\bar{y}_k}\}, Y_k^{HAS1} = \bar{y}_k\right) = 0$, then:

$$
\begin{aligned}
&= P\left(\{X_{k+1}^{HAS1} \in S_y\} | \{X_{k+1}^{HAS1} \in S_{\bar{y}_k}\}, Y_k^{HAS1} = \bar{y}_k\right) \cdot P(\{X_{k+1}^{HAS1} \in S_{\bar{y}_k}\} | \bar{Y}_k^{HAS1} = \bar{y}_k) \\
&= P(\bar{Y}_{k+1}^{HAS1} \leq y | Y_k^{HAS1} = \bar{y}_k).
\end{aligned}
$$

This proves condition (I).

We go on to prove (II), that $P(\bar{Y}_{k+1}^{HASE} \leq y | \bar{Y}_k^{HASE} = \bar{y}_k)$ is non-increasing in $\bar{y}_k$. Suppose that $\bar{y}_k$ and $\bar{y}'_k$ are such that $\bar{y}_k < \bar{y}'_k$. To show (II) we want to show that:

$$
P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k) \geq P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}'_k).
$$

The approach is to condition on the value of $\bar{y_k}^{high}$, and, since HASE samples on $S_{\bar{y_k}^{high}}$ in Step 2 of the algorithm, we know that $P(\bar{Y}_{k+1}^{HASE} \leq y | \bar{Y}_k^{HASE} = \bar{y}_k, \bar{y_k}^{high} = u)$ is non-increasing, we have:

$$P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k)$$

$$= \int_{-\infty}^{\infty} P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = z) \cdot dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k).$$

and because $\int_{-\infty}^{z} dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u) = P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = z) - P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = -\infty)$, and since $P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = -\infty) = 1$ (trivially), we substitute $P\left(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = z\right)$ as follows,

$$= \int_{-\infty}^{\infty} \left(1 + \int_{-\infty}^{z} dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)\right) \cdot dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k)$$

and reversing the order of integration, we get:

$$= 1 + \int_{-\infty}^{\infty} \int_{u}^{\infty} dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k) \cdot dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)$$

$$= 1 + \int_{-\infty}^{\infty} (1 - P(\bar{y}_k^{high} \leq u | Y_k^{HASE} = \bar{y}_k)) \cdot dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)$$

however, since $dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u) \leq 0$, since $P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)$ is non-increasing in $\bar{y}_k^{high}$, and since,

$$P(\bar{y}_k^{high} \leq u | Y_k^{HASE} = \bar{y}_k) \geq P(\bar{y}_k^{high} \leq u | Y_k^{HASE} = \bar{y}_k'), \tag{B.8}$$

the probability that $\bar{y}_k^{high}$ is low is always greater for $\bar{y}_k < \bar{y}_k'$, then

$$\geq 1 + \int_{-\infty}^{\infty} (1 - P(\bar{y}_k^{high} \leq u | Y_k^{HASE} = \bar{y}_k')) \cdot dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)$$

which is equivalent to

$$= 1 + \int_{-\infty}^{\infty} \int_{u}^{\infty} (dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k')) \cdot dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u)$$

and reversing the order of integration:

$$= 1 + \int_{-\infty}^{\infty} \int_{-\infty}^{z} dP(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = u) \cdot (dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k'))$$

$$= \int_{-\infty}^{\infty} P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = z) \cdot dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k')$$

therefore, since $P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k, \bar{y}_k^{high} = z) = P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k', \bar{y}_k^{high} = z)$, we write:

$$\geq \int_{-\infty}^{\infty} P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k', \bar{y}_k^{high} = z) \cdot dP(\bar{y}_k^{high} \leq z | Y_k^{HASE} = \bar{y}_k')$$

$$= P(\bar{Y}_{k+1}^{HASE} \leq y | Y_k^{HASE} = \bar{y}_k')$$

this demonstrates (II).

Lastly, we prove condition (III) from Lemma 11, by construction, both HAS1 and HAS-E sample on the entire domain so:

$$P(Y_0^{HASE} \leq y) = P(Y_0^{HAS1} \leq y)$$

so condition (III) is true trivially. This proves the theorem through reference to Lemma 11. □

Theorem 3: Given HAS-E then an upper bound on the expected number of iterations until reaching a value of $y_* + \varepsilon$ is:

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{d\rho(t)}{(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma \cdot p(t)}$$

$$= 1 + \frac{1}{(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma} \cdot ln\left(\frac{\nu(S)}{\nu(S_{y_*+\varepsilon})}\right) \tag{B.9}$$

Proof of Theorem 3

By stochastic dominance in Theorem 2, the expected number of iterations to achieving a value within $y_* + \varepsilon$ for HAS-E is less than or equal to the number for HAS1. Since the bettering probability for HAS1 is $b(y) = (1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma$ for all $y_* < y \leq y^*$, using (4.3), we have

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{d\rho(t)}{(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma \cdot p(t)}.$$

and since HAS1 uses uniform sampling, i.e., $p(y) = \frac{\nu(S_y)}{\nu(S)}$, we have

$$= 1 + \frac{1}{(1-\alpha) \cdot \left(\frac{\zeta_{low}}{\zeta_{high}}\right)^2 \cdot q \cdot \gamma} \cdot ln\left(\frac{\nu(S)}{\nu(S_{y_*+\varepsilon})}\right)$$

This proves the proposition. □

Lemma 12. *For a given constant a such that* $0 < a < 1$*, and a variable* $n \geq 1$*, then the function* $\frac{a^{1/n}}{1-a^{1/n}}$ *is bounded by a linear function of n, such that.*

$$f(n) = \frac{a^{1/n}}{1-a^{1/n}} \leq \frac{a}{1-a} + \frac{-log(a)}{(1-a)^2} \cdot n \tag{B.10}$$

Proof of Lemma 12

This bound is developed by proving that the derivative of $\frac{a^{1/n}}{1-a^{1/n}}$ obtains a maximum value over the range of n $in[1,\infty]$. Using this upper bound on the derivative, a linear function is determined to bound the expression. First, note that the if $n \geq 0$, the function is continuous and has defined derivatives. We take the first derivative of $f(n)$, yielding

$$f'(n) = \frac{df(n)}{dn} = \frac{-\frac{a^{1/n}log(a)}{n^2} \cdot (1-a^{1/n}) - \frac{a^{1/n}log(a)}{n^2} \cdot (a^{1/n})}{(1-a^{1/n})^2} = -\frac{a^{1/n} \cdot log(a)}{n^2 \cdot (1-a^{1/n})^2} \tag{B.11}$$

which is positive when $n > 0$. We go on to find a maximum value.

Examine part of the denominator in (B.11), let $\mathbf{d}(n) = n \cdot (1-a^{1/n})$ and we see that from Halley's Theorem:

$$\lim_{n\to\infty} n \cdot (1-a^{1/n}) = -log(a)$$

and at $n = 1$, then

$$\mathbf{d}(1) = (1-a)$$

We first examine the first derivative of $\mathbf{d}(n)$,

$$\mathbf{d}'(n) = \frac{d\mathbf{d}(n)}{dn} = -a^{\cdot 1/n} + \frac{a^{1/n} \cdot log(a)}{n} + 1$$

with $\lim_{n\to\infty} \mathbf{d}'(n) = 0$ and at $n = 1$, then

$$\mathbf{d}'(1) = -a + a \cdot log(a) + 1 = a \cdot (log(a) - 1) + 1. \tag{B.12}$$

Note that $\mathbf{d}'(1) > 0$ for $\forall a \in (-\infty, \infty)$ since (B.12) reaches a minimum in a of 0 at $a = 1$.

Next, we examine the second derivative of $\mathbf{d}(n)$,

$$\mathbf{d}''(n) = \frac{d^2\mathbf{d}(n)}{dn^2} = -\frac{a^{1/n} \cdot log^2(a)}{n^3}.$$

We note that $\mathbf{d}''(n) < 0$ for $0 < a < 1$ and $n > 1$. Since $0 > \mathbf{d}''(n)$ then $\mathbf{d}'(n) > 0$ is always positive since $\mathbf{d}'$ monotonically decreases from $a \cdot (log(a) - 1) + 1$ to 0 as n increases. Similarly, since $\mathbf{d}'(n) \geq 0$ for $n > 1$, then $\mathbf{d}(n)$ monotonically increases for $n > 1$. Therefore $\mathbf{d}(n)$ obtains a *minimum* at $n = 1$ therefore

$$n \cdot (1 - a^{1/n}) \geq (1 - a). \tag{B.13}$$

Returning to (B.11), we develop an upper bound since $-log(a)$ is positive and $a^{1/n} < 1$ for $0 < a < 1$,

$$f'(n) = -\frac{a^{1/n} \cdot log(a)}{n^2 \cdot (1 - a^{1/n})^2} \leq \frac{-log(a)}{\left(n \cdot (1 - a^{1/n})\right)^2}$$

then using (B.13),

$$f'(n) \leq \frac{-log(a)}{(1 - a)^2}.$$

Using an upper bound of the derivative, $f'(n)$, and the value of $f(1) = \frac{a}{1-a}$, an upper bound is determined for for $f(n)$ when $n \geq 1$ as

$$\frac{a^{1/n}}{1 - a^{1/n}} \leq \frac{a}{1 - a} + \frac{-log(a)}{(1 - a)^2} \cdot n.$$

which completes the proof. $\square$

B.3 Quantile Adaptive Search

Theorem 6: Consider the problem (P0). Let $\bar{Y}_k^{QAS}$ be the best sampled point at the kth iteration with parameters γ and $\mathscr{C}$. Let $\bar{Y}_k^{HAS2}$ be the best sampled of HAS2, the special case of HAS with bettering probability,b, equal to $\frac{\gamma}{\mathscr{C}} \cdot \left(\frac{\zeta_0^{low}}{\zeta_0^{high}}\right)^2$ and a sampling distribution ζ_0. $\bar{Y}_k^{QAS}$ stochastically dominates $\bar{Y}_k^{HAS2}$:

$$P(\bar{Y}_k^{QAS} \leq y) \geq P(\bar{Y}_k^{HAS2} \leq y) \text{ for } k = 0,1,2,\ldots,$$

where $y_* < y < y^*$.

Proof of Theorem 6

Similar to the proof for Theorem 2, we use the conditions provided in Lemma 11, that if

(I) $P(\bar{Y}_{k+1}^{QAS} \leq y | \bar{Y}_k^{QAS} = z) \geq P(\bar{Y}_{k+1}^{HAS2} | \bar{Y}_k^{HAS2} = z)$

(II) $P(\bar{Y}_{k+1}^{QAS} \leq y | \bar{Y}_k^{QAS} = z)$ is non-increasing in z, and

(III) $P(\bar{Y}_0^{QAS} \leq y) \geq P(\bar{Y}_0^{HAS2} \leq y)$

then $P(\bar{Y}_k^{QAS} \leq y) \geq P(\bar{Y}_k^{HAS2} \leq y)$ for $k = 0,1,2,\ldots,$ for $y < z$.

The first step is to demonstrate (I). To align notations with QAS let $\bar{y}_k = z$. We assume without loss of generality that $y < \bar{y}_k$ since otherwise $P(\bar{Y}_k^{QAS} \leq y) = P(\bar{Y}_k^{HAS2} \leq y) = 1$.
We rewrite the left-side of (I) in terms of the quantile level set,

$$P\left(\bar{Y}_{k+1}^{QAS} \leq y | \bar{Y}_k^{QAS} = \bar{y}_k\right) = P\left(\{X_{k+1}^{QAS} \in L(\delta_y, S)\} | \bar{Y}_k^{QAS} = \bar{y}_k\right). \tag{B.14}$$

where δ_y is the quantile level associated with y, such that $y = y(\delta_y, S)$.

$$\begin{aligned} &\geq P(\{X_{k+1}^{QAS} \in L(\delta_y, S)\} | \{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\}, \bar{Y}_k^{QAS} = \bar{y}_k) \\ &\cdot P(\{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\} | \bar{Y}_k^{QAS} = \bar{y}_k). \end{aligned} \tag{B.15}$$

and since $P\left(\{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\} | \bar{Y}_k^{QAS} = \bar{y}_k\right) = P\left(\{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\}\right)$, and by (ii) in Step2, in

QAS,

$$\geq P\left(\{X_{k+1}^{QAS} \in L(\delta_y, S)\} | \{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\}, \bar{Y}_k^{QAS} = \bar{y}_k\right) \cdot \gamma,$$

and using condition (iii) in Step 2, such that:

$$\begin{aligned}
&\gamma \cdot P\left(\{X_{k+1}^{QAS} \in L(\delta_y, S)\} | \{X_{k+1}^{QAS} \in L(\delta_{k+1}, S)\}, \bar{Y}_k^{QAS} = \bar{y}_k\right) \\
&\quad \geq \gamma \cdot P\left(\{X_0^{QAS} \in L(\delta_y, S)\} | \{X_0^{QAS} \in L(\delta_{k+1}, S)\}, \bar{Y}_k^{QAS} = \bar{y}_k\right) \\
&\quad = \gamma \cdot \frac{P(\{X_0^{QAS} \in L(\delta_y, S)\} \bigcap \{X_0^{QAS} \in L(\delta_{k+1}, S)\})}{P(\{X_0^{QAS} \in L(\delta_{k+1}, S)\})} \\
&\quad = \gamma \cdot \frac{P(\{X_0^{QAS} \in L(\delta_y, S)\})}{P(\{X_0^{QAS} \in L(\delta_{k+1}, S)\})}
\end{aligned}$$

and since $L(\delta_y, S) \subset L(\delta_{k+1}, S)$,

$$= \gamma \cdot \frac{\int_{L(\delta_y, S)} \zeta_0(x) \cdot dx}{\int_{L(\delta_{k+1}, S)} \zeta_0(x) \cdot dx}$$

and since $\zeta_0^{low} \leq \zeta_0(x) \leq \zeta_0^{high}$:

$$= \gamma \cdot \frac{\int_{L(\delta_y, S)} \zeta_0(x) \cdot dx}{\int_{L(\delta_{k+1}, S)} \zeta_0(x) \cdot dx} \geq \gamma \cdot \frac{\zeta_0^{low} \cdot \int_{L(\delta_y, S)} dx}{\zeta_0^{high} \cdot \int_{L(\delta_{k+1}, S)} dx} = \gamma \cdot \frac{\zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\zeta_0^{high} \cdot \nu(L(\delta_{k+1}, S))}.$$

Therefore based on the ratio in condition *(i)*:

$$= \gamma \cdot \frac{\zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\zeta_0^{high} \cdot \delta_{k+1} \cdot \nu(S)} \geq \gamma \cdot \frac{\zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\zeta_0^{high} \cdot \mathscr{C} \cdot \delta_{(k)} \cdot \nu(S)} = \gamma \cdot \frac{\zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\mathscr{C} \cdot \zeta_0^{high} \cdot \nu(L(\delta_{(k)}, S))}.$$

To summarize thus far, we have a lower-bound on the left-hand side of (I)

$$P\left(\bar{Y}_{k+1}^{QAS} \leq y | \bar{Y}_k^{QAS} = \bar{y}_k\right) \geq \frac{\gamma \cdot \zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\mathscr{C} \cdot \zeta_0^{high} \cdot \nu(L(\delta_{(k)}, S))}. \tag{B.16}$$

Now, rewrite the right-hand side of (I)

$$P(\bar{Y}_{k+1}^{HAS2} \leq y | \bar{Y}_k^{HAS2} = \bar{y}_k)$$

$$= P(\{X_{k+1}^{HAS2} \in L(\delta_y, S)\} | \{X_k^{HAS2} \in L(\delta_{(k)}, S)\}, \bar{Y}_k^{HAS2} = \bar{y}_k)$$

$$\cdot P(\{X_{k+1}^{HAS2} \in L(\delta_{(k)}, S)\} | \bar{Y}_k^{HAS2} = \bar{y}_k)$$

$$= b \cdot \frac{\int_{L(\delta_y,S)} \zeta_0(x) \cdot dx}{\int_{L(\delta_{(k)},S)} \zeta_0(x) \cdot dx} \leq b \cdot \frac{\zeta_0^{high} \cdot \nu(L(\delta_y, S))}{\zeta_0^{low} \cdot \nu(L(\delta_{(k)}, S))}$$

which results in,

$$P(\bar{Y}_{k+1}^{HAS2} \leq y | \bar{Y}_k^{HAS2} = \bar{y}_k) \leq b \cdot \frac{\zeta_0^{high} \cdot \nu(L(\delta_y, S))}{\zeta_0^{low} \cdot \nu(L(\delta_{(k)}, S))} \tag{B.17}$$

therefore combining (B.16) and (B.17) yields,

$$P(\bar{Y}_k^{QAS} \leq y | \bar{Y}_k^{QAS} = \bar{y}_k) \geq \left(\frac{\gamma}{\mathscr{C}}\right) \cdot \frac{\zeta_0^{low} \cdot \nu(L(\delta_y, S))}{\zeta_0^{high} \cdot \nu(L(\delta_{(k)}, S))}$$

$$= \left(\frac{\gamma}{\mathscr{C}}\right) \cdot \left(\frac{\zeta_0^{low} \cdot \zeta_0^{low} \cdot \zeta_0^{high} \cdot \nu(L(\delta_y, S))}{\zeta_0^{high} \cdot \zeta_0^{high} \cdot \zeta_0^{low} \cdot \nu(L(\delta_{(k)}, S))} \right)$$

$$= \left(\frac{\gamma}{\mathscr{C}}\right) \cdot \left(\frac{(\zeta_0^{low})^2 \cdot \zeta_0^{high} \cdot \nu(L(\delta_y, S))}{(\zeta_0^{high})^2 \cdot \zeta_0^{low} \cdot \nu(L(\delta_{(k)}, S))} \right)$$

and since the bettering probability $\frac{\gamma}{\mathscr{C}} \cdot \left(\frac{\zeta_0^{low}}{\zeta_0^{high}}\right)^2 = b$, and from (B.17)

$$= b \cdot \frac{\zeta_0^{high} \cdot \nu(L(\delta_y, S))}{\zeta_0^{low} \cdot \nu(L(\delta_{(k)}, S))} \geq P\left(\bar{Y}_k^{HAS2} \leq y | \bar{Y}_k^{HAS2} = \bar{y}_k\right).$$

Therefore condition (I) is proved for all $k = 0, 1, 2, \ldots$

Condition (II) holds based explicitly on condition (iv). Moreover, for Condition (III), holds by construction both *HAS*2 and *QAS*, which sample according to the distribution ζ_0 on the first iteration,

$$P(Y_0^{QAS} \leq y) = P(Y_0^{HAS2} \leq y)$$

Therefore, the proposition is demonstrated by Lemma 11. □

Theorem 7: The expected number of iterations until the value of $y_* + \varepsilon$ or better is sampled by the

QAS algorithm can be given an upper bound of:

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{\mathscr{C}}{\gamma} \cdot \left(\frac{\zeta_0^{high}}{\zeta_0^{low}} \right)^2 \cdot \left(\frac{d\rho(t)}{p(t)} \right)$$

Proof of Theorem 7

By stochastic dominance in Theorem 6, the expected number of iteration to achieving a value within $y_* + \varepsilon$ for QAS is less than or equal to the number for HAS2. Since the better probability for HAS2 is $b(y) = \frac{\gamma}{\mathscr{C}} \cdot \left(\frac{\zeta_0^{low}}{\zeta_0^{high}} \right)^2$, for $y_* < y \leq y^*$, using (4.3), we have

$$E(N(y_* + \varepsilon)) \leq 1 + \int_{y_*+\varepsilon}^{\infty} \frac{\mathscr{C}}{\gamma} \cdot \left(\frac{\zeta_0^{high}}{\zeta_0^{low}} \right)^2 \cdot \left(\frac{d\rho(t)}{p(t)} \right)$$

This leads to an upper-bound for *QAS* written in (4.13). □

B.4 Quantile Nested Partitions

We can write the proof of Theorem 9, that generally extend the theorems proving the convergence of the Nested Partition algorithm.

Theorem 9 As $k \to \infty$ then $P\left(\sigma^B(k) = \sigma^*\right) \to 1$.

Proof of Theorem 9

On every iteration, the algorithm either branches the most promising region $\sigma^B(k)$, or the most promising region is unbranchable in which case $\sigma^B(k) \in \Sigma_{max}$ and $\hat{y}\left(\delta, \sigma^B(k)\right) < \hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j)}, \sigma_j^k, \right)$ for all $\sigma_j^k \in \Sigma_{contend}(k)$ and $\sigma_j^k \notin \Sigma_{max}$.

As $k \to \infty$ then $v_{min}(k) \to v(\sigma^*)$ and the number of samples $N_j^k \to \infty$ in every σ_j^k in $\Sigma_{contend}(k)$, therefore

$$\hat{y}\left(\delta \cdot \frac{\mathbf{v}_{min}(k)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k \right) \to y\left(\delta \cdot \frac{\mathbf{v}(\sigma^*)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k \right)$$

with probability arbitrarily close to 1 by the consistency of the quantile estimator [91, 23].

Due to the consistency of the estimator and since the number of possible regions in $\Sigma_{contend}(k)$ is finite, as $k \to \infty$, it is true with probability approaching 1 that for any region $\sigma_j^k \in \Sigma_{contend}(k)$ either

$\sigma_j^k \in \Sigma_{max}$ (unbranchable) or $\exists\ \bar{\sigma}_j^k \in \Sigma_{max} \bigcap \Sigma_{contend}(k)$ such that $y\left(\delta, \bar{\sigma}_j^k(k)\right) < y\left(\delta \cdot \frac{\mathbf{v}(\sigma^*)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right)$ and $\sigma^* \in \Sigma_{contend}(k)$.

To see this, consider two regions, $\sigma_j, \sigma_{j'}$ such that $\sigma_j \subset \sigma_{j'}$. Then $y(\delta \cdot \frac{\mathbf{v}(\sigma_j)}{\mathbf{v}(\sigma_{j'})}, \sigma_{j'}) \leq y(\delta, \sigma_j)$ since the set of points $\{x : f(x) < y(\delta, \sigma_j)\}$ is also contained in σ_j and therefore constitutes at least $\delta \cdot \frac{\mathbf{v}(\sigma_j)}{\mathbf{v}(\sigma_{j'})}$ of the total volume of σ_j'. Now, if for all regions $\sigma_j^k \in \Sigma_{contend}(k)$ either $\sigma_j^k \in \Sigma_{max}$ (unbranchable) or $\exists \bar{\sigma}_j^k \in \Sigma_{max} \bigcap \Sigma_{contend}(k)$ such that $y\left(\delta, \bar{\sigma}_j^k(k)\right) < y\left(\delta \cdot \frac{\mathbf{v}(\sigma^*)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k, \right)$ then for any branchable region that contains σ^*, i.e., $\sigma^* \subset \tilde{\sigma}_j^k$, we have $y(\delta \cdot \frac{\mathbf{v}(\sigma^*)}{\mathbf{v}(\tilde{\sigma}_j^k)}, \tilde{\sigma}_j^k) \leq y(\delta, \sigma^*) < y(\delta, \bar{\sigma}_j^k)$ for all $\bar{\sigma}_j{}^k \in \Sigma_{max} \bigcap \Sigma_{contend}(k)$ and therefore $\tilde{\sigma}_j^k \notin \Sigma_{contend}(k)$. Since at least one region in $\Sigma_{contend}(k)$ contains σ^* then $\sigma^* \in \Sigma_{contend}(k)$.

Therefore as $k \to \infty$ the probability that $\sigma^* \in \Sigma_{contend}(k)$ and $y(\delta, \sigma^*) < y\left(\delta \cdot \frac{\mathbf{v}(\sigma^*)}{\mathbf{v}(\sigma_j^k)}, \sigma_j^k\right)\ \ \forall \sigma_j^k \in \Sigma_{contend}(k)$ goes to 1, and as $k \to \infty$ then $P(\hat{y}(\delta, \sigma^*) < \hat{y}(\delta, \sigma^m(k))) \to 1$ and the probability $P\left(\sigma^B(k) = \sigma^*\right) \to 1$. $\square$

Theorem 10 If f is a function that satisfies the Lipschitz condition with Lipschitz constant L, there exists a value δ^* such that for all $\delta < \delta^*$ then P($x^* \in \sigma^B(k)) \to 1$ as $k \to \infty$.

Proof of Theorem 10

Consider $\sigma_i \in \Sigma_{max}$ for $i = 1, \ldots, \|\Sigma_{max}\|$. Let $x_i^* = argmin_{x \in \sigma_i} f(x)$. Order the unbranchable regions in Σ_{max} by their minimum values, i.e., $\sigma_{(1)}, \cdots, \sigma_{(\|\Sigma_{max}\|)}$ such that $f(x_{(1)}^*) \leq \cdots \leq f(x_{(\|\Sigma_{max}\|)}^*)$.

Consider $\sigma_{(1)}$ and $\sigma_{(2)}$ with $f(x_{(1)}^*) < f(x_{(2)}^*)$. From the Lipschitz condition there must be a hypersphere $\mathbf{h}$ centered at $x_{(1)}^*$ and $\mathbf{v}\left(\mathbf{h} \bigcap \sigma_{(1)}\right) > 0$, such that $\forall x \in \mathbf{h} : f(x) < f(x_{(2)}^*)$. Therefore for $\delta < \delta^* = \frac{\mathbf{v}(\mathbf{h} \bigcap \sigma_{(1)})}{\mathbf{v}(\sigma_{(1)})}$, we are assured that $y(\delta, \sigma_{(1)}) < y(\delta, \sigma_{(i)})\ \ \forall i \neq 1$. The best σ^* relative to a $\delta < \delta^*$ satisfies $y(\delta, \sigma^*) \leq y(\delta, \sigma_{(i)})$ for all i, and therefore $\sigma^* = \sigma_{(1)}$ and $x^* \in \sigma^*$.

By Theorem 9, $P\left(\sigma^B(k) = \sigma^*\right) \to 1$ as $k \to \infty$. Therefore $P\left(x^* \in \sigma^B(k)\right) \to 1$ as $k \to \infty$. $\square$

Appendix C

TABLES AND PARAMETERS FOR OPTIMIZATION

C.1 Staffing Model

Table C.1: Fit Parameters for Normal and Weibull models

clinic	modeled normal mean : $\hat{\mu}_{c,i}$	modeled normal standard deviation : $\hat{\sigma}_{c,i}$	modeled Weibull shape : $\hat{\alpha}_{c,i}$	modeled weibull scale : $\hat{\beta}_{c,i}$
1	4553, 3672, 2741	1056, 854, 644	5.19, 4.99, 4.95	5048, 4007, 3008
2	1611, 973, 640	1669, 943, 629	1.02, 0.99, 1.00	989.32, 655, 636
3	1512, 1003, 619	1479, 1013, 596	1.01, 1.01, 1.01	1577, 1020, 671
4	1600, 876, 703	3866, 1852, 1723	0.51, 0.53, 0.50	759.08, 539, 305
5	1529, 1082, 595	2996, 2683, 1287	0.47, 0.50, 0.494	715, 458, 328
6	4578, 3664, 2282	1062, 874, 535	4.89, 4.98, 5.07	5032, 3978, 2507
7	1594, 938, 632	3660, 2031, 1526	0.50, 0.52, 0.51	706, 563, 318

Table C.2: System Parameters

γ_i	K	ξ	w_i
0.05	1000	0.0001	1, 2, 2

Table C.3: Co-morbidity matrix $m_{i,'i}$

	specialist type 1	specialist type 2	specialist type 3
specialist type 1	1	0	0
specialist type 2	0.1	0.9	0
specialist type 3	0.1	0.1	0.8

Table C.4: Transport Penalty : $t_{c,l}$

	clinic 1	**clinic 2**	**clinic 3**	**clinic 4**	**clinic 5**	**clinic 6**	**clinic 7**
clinic 1	0	0.5, 1, 1	3, 6, 6	3, 6, 6	3.5, 7, 7	3.5, 7, 7	2, 4, 4
clinic 2	0.5, 1, 1	0	3, 6, 6	2.5, 5, 5	3, 6, 6	3, 6, 6	4, 8, 8
clinic 3	3, 6, 6	3, 6, 6	0	6, 12, 12	6.5, 13, 13,	7, 14, 14,	8, 16, 16,
clinic 4	3, 6, 6	2.5, 5, 5	6, 12, 12	0	3, 6, 6	3, 6, 6	2.5, 5, 5 10, 10
clinic 5	3.5, 7, 7	3, 6, 6	6.5, 13, 13	1.5, 3, 3	0	1.5, 3, 3	5.5, 11, 11
clinic 6	3.5, 7, 7	3, 6, 6	7, 14, 14	1.5, 3, 3	1.5, 3, 3	0	6.5, 11, 11
clinic 7	2, 4, 4	4, 8, 8	8, 16, 16	5, 10, 10	5.5, 11, 11	5.5, 11, 11	0

Table C.5: Clinic Data

clinic	**capacity :** $b_{l,i}$	**cost per staff :** $h_{l,i}$	**discontinuity penalty rate :** $s_{l,i}$	**discontinuity threshold :** $v_{l,i}$
1	300000	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
2	3000, 2000, 1300	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
3	3000, 2000 1300	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
4	1500, 1000, 600	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
5	1500, 1000, 600	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
6	10000, 8000, 5000	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300
7	1500, 1000, 600	2, 1, 0.75	0.5, 0.3, 1	3000, 2000, 1300

Values generated for $(d_{c,i,1}, d_{c,i,2}, ..., d_{c,i,K})$ can be provided upon request.

C.2 Optimal Patient Paneling Model

Indices	
J	Number of physician panels, indexed by $j = 1,\ldots,J$
L	Number of clinic locations, indexed by $l = 1,\ldots,L$
R	Regions, indexed by $r = 1,\ldots,R$
M	Morbidity categories indexed by $b = 1,\ldots,M$
I	Number of patient populations, indexed by $i = 1,\ldots,I$
Decision Variables	
$X_{i,j}$	the percentage of patients from population i being assigned to physician panel j
Y_j	an integer value taking values $\{1,\ldots,L\}$ indicating that we are locating physician j at location l
V_j	a continuous variable indicating the amount of virtual care time available to panel j
Parameters	
$d_{r,l}$	the distance between population at region r and clinic l
v_i	the base likelihood that patient type i will use virtual
o_i	the base likelihood that patient type i will scheduled (as opposed to same-day arrival)
$n_{m,r,l}$	probability of a no-show for morbidity type m in region r when seeking care at clinic l
b_i	the maximum watitime before balk for patient type i
h_j	hours of availability for physician in panel j
$t_{frameLimit}$	the timeframe of the scheduling
f_j	The preferred utilization of physician j
v_j^{low}, v_j^{up}	The upper and lower bounds for the available virtual hours at panel j
λ_i	Average arrival time for patient group i for designated time period
$\chi_{m,r}$	the percent morbidity type m at region r
Objective Function Weights	
ω_i^{defer}	Average arrival time for patient group i for designated time period
$\omega_i^{waittime}$	The objective function weight for waittime for patient type i
ω_i^{noshow}	The objective function weight for waittime for patient type i
ω_i^{travel}	The objective function weight for waittime for patient type i
ω_i^{util}	The objective function weight for utilization for panel type j
Output Metrics	
T_i	The travel time for patient p_i
W_i	The wait time for a patient p_i
B_i	A indicator variables whether patient p_i defers care or experiences extreme wait time causing them to balk
N_i	A indicator variables whether patient p_i is a no-show
S_i	A indicator variables whether patient p_i is scheduled as opposed to virtual
U_j	The average number of hours utilized for Physician j

Table C.6: System Indices, Parameters, Decision Variables, and Random Variable Parameters for the model

Table C.7: The arrival rate break downs based on group health data across four RUB categories (grouped into two categories)

Average RUB (rounded)	Avg Visits (per year)	number patients	number visits	number of visits (2 categories)	number of patients (2 categories)	average yearly visits
0	0.89	33356.00	29597.71	199584.00	106873.00	1.87
1	2.31	73517.00	169986.29			
2	4.14	101805.00	421249.19	479665.65	108587.00	4.42
3	8.61	6782.00	58416.46			

Table C.8: Arrival time between arrivals for each of the broad rub categories

	The two RUB Model	Avervage yearly arrival	Total yearly arrival	Average time between arrivals (hours)
RUB1	0.50	1.87	24081.25	0.015
RUB2	0.50	4.42	57871.57	0.006

Table C.9: The simulation model's inputs concerning arrival rate, (base) no-show probability, and the percentage of virtual care usage

	Arrival Rate	Schedule Probability	No Show Probability	Virtual Care probability
RUB 1	0.015	0.9	0.1	0.15
RUB 2	0.006	0.9	0.05	0.02

Table C.10: The travel weights used for objective function measurement

	Clinic 1	Clinic 2	Clinic 3
Location 1	0	0.33	1
Location 2	0.33	0	1
Location 3	1	1	0

Table C.11: Input variables relative to each of the three locations selected. The percentage of each of the morbidity categories included in the simulation model

	% RUB1 Population at Location	% RUB2 Population at Location
Location 1	30 %	25 %
Location 2	50 %	30 %
Location 3	20 %	45 %
Total	100 %	100 %

Table C.12: Input variables relative to each of the three locations selected. The percentage of each of the morbidity categories included in the simulation model

	% Location demand from RUB 1	**% Location demand from RUB 1**	**Toal**
Region 1	54%	46%	100
Region 2	62%	38%	100
Region 3	30%	70%	100

Table C.13: The objective function weights used for in the objective-function measurements

	Objective weight for Balks	Objective weight for NoShows	Objective weight for travel time	Objective weight for clinic wait time	Objective weight for virtual wait time
RUB 1, m=1	1	1	1	1	1
RUB 2, m=2	2	2	2	2	1